Programme Name : BTech-ICT

Semester : 4

Course Name : CSE250 Database Management System

Project title : Hospital Management System

Group No : 25

Group Members :

AU1841058	**Deepkumar Patel**
AU1841081	**Dhruv Panchal**
AU1841084	**Shivam Lakhtariya**

● **Description of Project**

Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep
the hospital running smoothly & successfully.

Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as "Hospital Management System"

The main aim of our project is to provide a paperless hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

● **Functionalities**

Our hospital management system contains many modules like a login module, patient module, admin module, doctor module, billing module ,etc.

1. Login module

Basically, the login module records the username and password of any patients, doctors ,and admin.

- Login patient
- Login doctor
- Login admin

2. Patient Module

The patient module keeps track of all details about both in-patient and out-patient. It contains all the details of the patient like patient id, patient name, address, appointments, booking date, doctor name, room no, etc.
A patient can book appointments with any type of doctors like MBBS or Ms.
Also, patients can see booked appointments or the status of appointments.

- Book appointment
- Show appointment
- Edit profile
- Medical history
- Forgot password
- Update details
- Add complain

3. Admin module

This module handles all the master entry details of the hospital requirement such as doctor specialization, appointment fee and any types of service charges. Admin can modify patient and doctor details like Add doctor, update doctor, remove doctor, Add patient, update patient, remove patient.

- Add doctor
- Update doctor
- Remove doctor
- Search doctor
- Add patient
- Update patient
- Remove patient
- Search patient
- View complain
- Add room

4. Doctor module

In this module, doctors can see all the booked appointments of patients.
And doctors can modify appointments either accept appointments or reject appointments.

- View appointment
- Edit profile
- Admit patient
- Modify appointment
- Patients report

4. Billing module

This Module generates the final bill of patients. It also includes payment details of patients.

- Generate bill report
- Payment method

- ## Software Specification

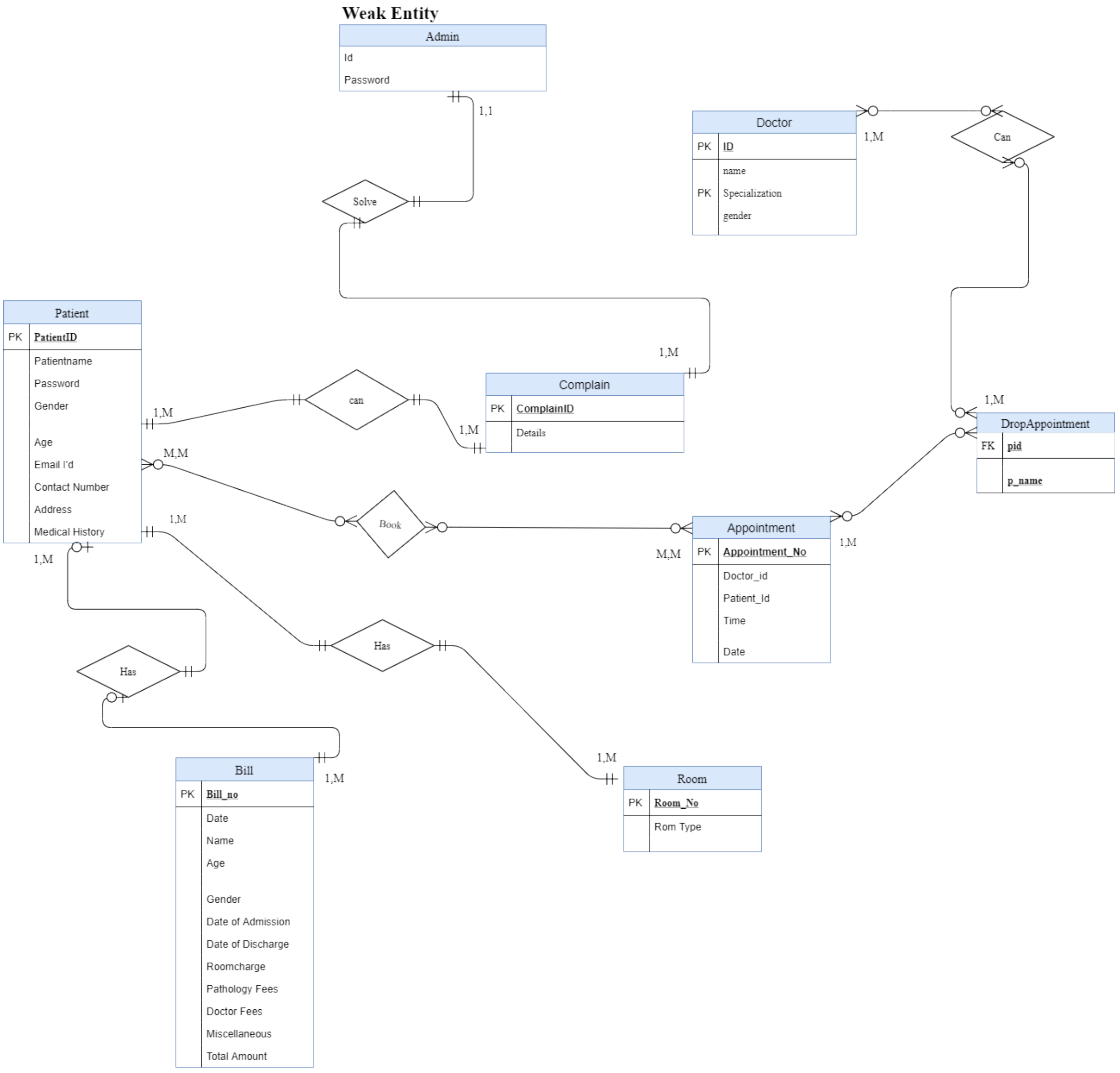
1. **Oracle 11g**

   Basically oracle Database is a relational database.Also it has object and extensible Markup Language capabilities. Main reliability of an oracle database is that it enables you to store data, update data, and retrieve data. We used oracle 11g because it has high performance and reliability.

2. **Java**

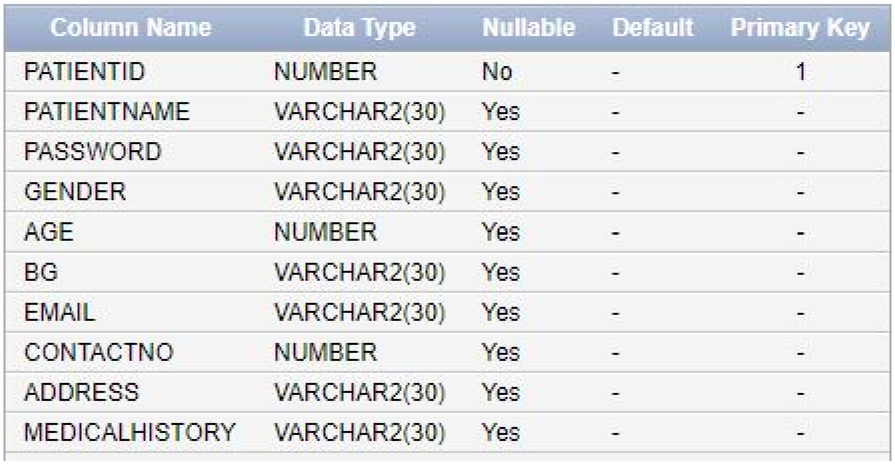
   Java is the object oriented programming language and also java contains GUI(Graphics user interface). We selected Java for our project because it's easy to interface an Oracle database and it's reliable to users.

- **E-R Diagram**



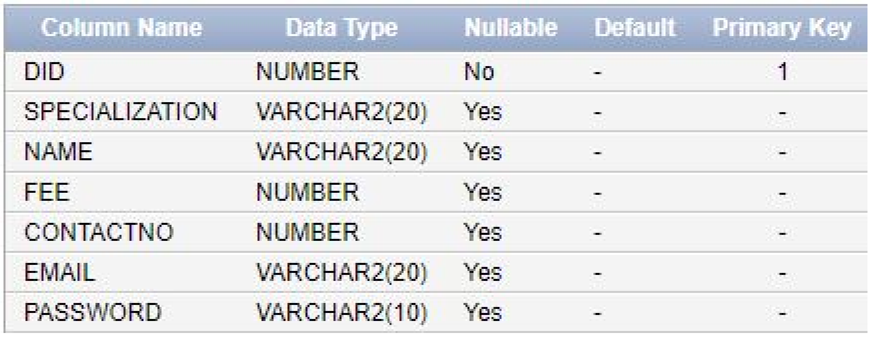
● Table Design

- **Patient Table**



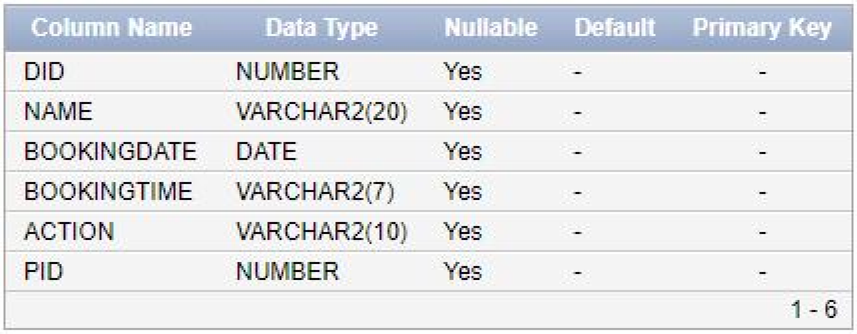
Column Name	Data Type	Nullable	Default	Primary Key
PATIENTID	NUMBER	No	-	1
PATIENTNAME	VARCHAR2(30)	Yes	-	-
PASSWORD	VARCHAR2(30)	Yes	-	-
GENDER	VARCHAR2(30)	Yes	-	-
AGE	NUMBER	Yes	-	-
BG	VARCHAR2(30)	Yes	-	-
EMAIL	VARCHAR2(30)	Yes	-	-
CONTACTNO	NUMBER	Yes	-	-
ADDRESS	VARCHAR2(30)	Yes	-	-
MEDICALHISTORY	VARCHAR2(30)	Yes	-	-

- **Doctor Table**



Column Name	Data Type	Nullable	Default	Primary Key
DID	NUMBER	No	-	1
SPECIALIZATION	VARCHAR2(20)	Yes	-	-
NAME	VARCHAR2(20)	Yes	-	-
FEE	NUMBER	Yes	-	-
CONTACTNO	NUMBER	Yes	-	-
EMAIL	VARCHAR2(20)	Yes	-	-
PASSWORD	VARCHAR2(10)	Yes	-	-

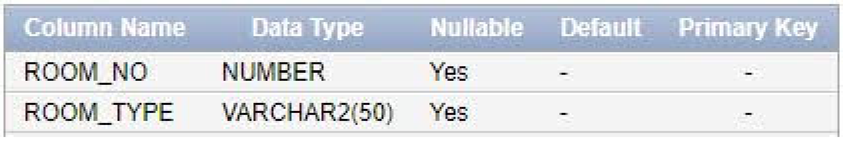
- **Appointments Table**



Column Name	Data Type	Nullable	Default	Primary Key
DID	NUMBER	Yes	-	-
NAME	VARCHAR2(20)	Yes	-	-
BOOKINGDATE	DATE	Yes	-	-
BOOKINGTIME	VARCHAR2(7)	Yes	-	-
ACTION	VARCHAR2(10)	Yes	-	-
PID	NUMBER	Yes	-	-

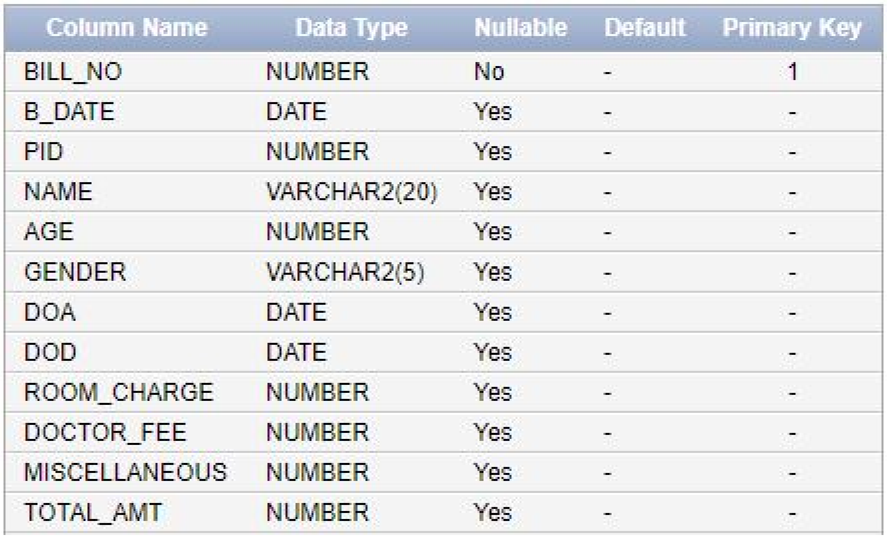
1 - 6

- **Room Table**



Column Name	Data Type	Nullable	Default	Primary Key
ROOM_NO	NUMBER	Yes	-	-
ROOM_TYPE	VARCHAR2(50)	Yes	-	-

- **Bill Table**



Column Name	Data Type	Nullable	Default	Primary Key
BILL_NO	NUMBER	No	-	1
B_DATE	DATE	Yes	-	-
PID	NUMBER	Yes	-	-
NAME	VARCHAR2(20)	Yes	-	-
AGE	NUMBER	Yes	-	-
GENDER	VARCHAR2(5)	Yes	-	-
DOA	DATE	Yes	-	-
DOD	DATE	Yes	-	-
ROOM_CHARGE	NUMBER	Yes	-	-
DOCTOR_FEE	NUMBER	Yes	-	-
MISCELLANEOUS	NUMBER	Yes	-	-
TOTAL_AMT	NUMBER	Yes	-	-

- **Procedures**

- **INSERT PATIENT PROCEDURE**

```
create or replace procedure insertpatient(p_id
patients.patientid%type,p_name patients.patientname%type,p_password
patients.password%type,p_gender patients.gender%type,p_age
patients.age%type,p_bg patients.bg%type,p_email
patients.email%type,p_contactno patients.contactno%type,p_address
patients.address%type,p_medical patients.medicalhistory%type) as

begin
   insert into patients
values(p_id,p_name,p_password,p_gender,p_age,p_bg,p_email,p_contactno,p_a
ddress,p_medical);
end;
```

<u>Procedure call</u>

```
CallableStatement cStmt = conn.prepareCall("{call
insertpatient(?,?,?,?,?,?,?,?,?)}");
cStmt.setInt(1, pid);
cStmt.setString(2,pname);
cStmt.setString(3, password);
cStmt.setString(4,gender);
cStmt.setString(5, age);
cStmt.setString(6, blood group);
cStmt.setSInt(7,contactno);
cStmt.setString(8, address);
cStmt.setString(9, medicalhistory);
```

- **DELETE PATIENT PROCEDURE**

```
create or replace procedure deletepatient(p_id patients.patientid%type)
as
begin
   delete from patients where patientid = p_id;
end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call deletepatient(?)}");
cStmt.setInt(1, pid);
```

- **UPDATE PATIENT PROCEDURE**

```
create or replace procedure updatepatient(id int,p_id
patients.patientid%type,p_name patients.patientname%type,p_password
patients.password%type,p_gender patients.gender%type,p_age
patients.age%type,p_bg patients.bg%type,p_email
patients.email%type,p_contactno patients.contactno%type,p_address
patients.address%type,p_medical patients.medicalhistory%type) as

begin
   update patients set PatientID = p_id, Patientname = p_name ,Password =
p_password ,Gender=p_gender ,
Age = p_age , Bg = p_Bg, Email = p_Email , ContactNo = p_ContactNo,
Address = p_Address, Medicalhistory = p_Medical where patientid = id;

end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call
updatepatient(?,?,?,?,?,?,?,?,?)}");
cStmt.setInt(1, pid);
cStmt.setString(2,pname);
cStmt.setString(3, password);
cStmt.setString(4,gender);
cStmt.setString(5, age);
cStmt.setString(6, blood group);
cStmt.setSInt(7,contactno);
cStmt.setString(8, address);
cStmt.setString(9, medicalhistory);
```

- **INSERT DOCTOR PROCEDURE**

```
create or replace insertdoctor(d_id doctor.did%type,d_sp
doctor.specialization%type,d_name doctor.name%type,d_fee
doctor.fee%type,d_contactno doctor.contactno%type,d_email
doctor.email%type
,d_pass doctor.password%type) as
begin
insert into doctor
values(d_id,d_sp,d_name,d_fee,d_contactno,d_email,d_pass);
end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call
insertdoctor(?,?,?,?,?,?,?)}");
cStmt.setInt(1, did);
cStmt.setString(2,specialization);
cStmt.setString(3,dname);
cStmt.setInt(4,fee);
cStmt.setInt(5, contactno);
cStmt.setString(6, email);
cStmt.setSInt(7,password);
```

- **UPDATE DOCTOR PROCEDURE**

```
create or replace procedure updatedoctor(idd int,d_id
doctor.did%type,d_sp doctor.specialization%type,d_name
doctor.name%type,d_fee doctor.fee%type,d_contactno
doctor.contactno%type,d_email doctor.email%type
,d_pass doctor.password%type) as

begin
   update doctor set did = d_id,specialization=d_sp
,name=d_name,fee=d_fee,contactno=d_contactno,email=d_email,password =
d_pass where did = idd;

end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call
updatedoctor(?,?,?,?,?,?,?)}");
cStmt.setInt(1, did);
cStmt.setString(2,specialization);
cStmt.setString(3,dname);
cStmt.setInt(4,fee);
cStmt.setInt(5, contactno);
cStmt.setString(6, email);
cStmt.setSInt(7,password);
```

- ## DELETE DOCTOR PROCEDURE

```
create or replace procedure deletedoctor(d_id doctor.did%type) as
begin
   delete from doctor where did = d_id;
end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call deletedoctor(?)}");
cStmt.setInt(1, did);
```

- **BOOK APPOINTMENT**

```
create or replace procedure book_ap(d1 int,n varchar,d date,t varchar,a
varchar,d2 int) as
begin
INSERT INTO
book_appointment(did,name,bookingdate,bookingtime,action,pid)
VALUES(d1,n,d,t,a,d2);
end;
```

Procedure call

```
 CallableStatement cStmt = conn.prepareCall("{call
book_ap(?,?,?,?,?,?)}");
cStmt.setInt(1, rs3.getInt(1));
cStmt.setString(2,jComboBox2.getSelectedItem().toString());
cStmt.setString(3, d);
cStmt.setString(4,jTextField1.getText());
cStmt.setString(5, "Booked");
cStmt.setInt(6, rs4.getInt(1));
```

- **Add Bill**

```
  create or replace insertbill(b_id BillDetails.bill_no%type,b_date
BillDetails.Bill_date%type,P_ids BillDetails.P_id%type,b_name
BillDetails.name%type,b_age BillDetails.Age%type,b_gen
BillDetails.Gender%type,b_adm BillDetails.Date_admission%type,b_dis
BillDetails.Date_Discharge%type,b_roomc
BillDetails.Room_Charge%type,b_pathc
BillDetails.Pathology_fees%type,b_doctorc
BillDetails.Doctor_Fees%type,b_mcharge
BillDetails.Miscellaneous_Charge%type,b_total
BillDetails.Total_Amount%type) as

 Begin

 insert into insertbill
 values(b_id,b_date,p_ids,b_name,b_age,b_gen,b_adm,b_dis,b_roomc,b_pathc,b
 _doctorc,b_mcharge,b_total);

 end;
```

- **UPDATE Bill PROCEDURE**

```
create or replace procedure updatedoctor(b_id
BillDetails.bill_no%type,b_date BillDetails.Bill_date%type,P_ids
BillDetails.P_id%type,b_name BillDetails.name%type,b_age
BillDetails.Age%type,b_gen BillDetails.Gender%type,b_adm
BillDetails.Date_admission%type,b_dis
BillDetails.Date_Discharge%type,b_roomc
BillDetails.Room_Charge%type,b_pathc
BillDetails.Pathology_fees%type,b_doctorc
BillDetails.Doctor_Fees%type,b_mcharge
BillDetails.Miscellaneous_Charge%type,b_total
BillDetails.Total_Amount%type) as

begin
   update bill set (Bill_no = d_id,Bill_date = d_sp ,P_id = d_name,name =
d_fee,Age=d_contactno,Gender= ,Date_admission=d_email,Date_Discharge =
d_pass,Room_Charge where did = idd;

end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call
updatedoctor(?,?,?,?,?,?,?)}");
cStmt.setInt(1, did);
cStmt.setString(2,specialization);
cStmt.setString(3,dname);
cStmt.setInt(4,fee);
cStmt.setInt(5, contactno);
cStmt.setString(6, email);
cStmt.setSInt(7,password);
```

- **DELETE Bill PROCEDURE**

```
create or replace procedure deletebill(b_id bill.did%type) as
begin
   delete from doctor where did = d_id;
end;
```

Procedure call

```
CallableStatement cStmt = conn.prepareCall("{call deletebill(?)}");
cStmt.setInt(1, did);
```

- ## Triggers

- **To save deleted records**

```
create or replace trigger tr_delete after delete on patients
for each row

Begin

  insert into deletedpatients values(:old.PatientID, :old.Patientname
,:old.Password,:old.Gender , :old.Age , :old.Bg , :old.Email ,
:old.ContactNo , :old.Address , :old.Medicalhistory);

end;
```

- **For Auto increment of doctor id**

```
CREATE SEQUENCE SYSTEM.MYSEQ
  START WITH 1
  MAXVALUE 999999999999999999999999999
  MINVALUE 1
  NOCYCLE
  CACHE 20
  NOORDER;

CREATE OR REPLACE TRIGGER TR_CITY  BEFORE INSERT   ON doctor FOR EACH ROW
BEGIN   SELECT LPAD(LTRIM(RTRIM(TO_CHAR(myseq.NEXTVAL))),10,'0')   INTO
:NEW.id  FROM DUAL;
END;
```

- **For Auto increment of patientid**

```
CREATE SEQUENCE SYSTEM.MYSEQ
  START WITH 1
  MAXVALUE 999999999999999999999999999
  MINVALUE 1
  NOCYCLE
  CACHE 20
  NOORDER;
```

```
CREATE OR REPLACE TRIGGER TR_CITY  BEFORE INSERT   ON patients FOR EACH
ROW
BEGIN   SELECT LPAD(LTRIM(RTRIM(TO_CHAR(myseq.NEXTVAL))),10,'0')   INTO
:NEW.id  FROM DUAL;
END;
```

- **After insertion into patient**

```
create or replace trigger tr_insert after insert on patients
for each row
begin

if inserting then
  DBMS_OUTPUT.PUT_LINE( 'Inserted successfully' );
else
raise_application_error(-20001,'Error occurred while inserting.');
end if;

end;
```

- **After updating into patient**

```
create or replace trigger tr_update after update on patients
for each row
begin

if updating then
  DBMS_OUTPUT.PUT_LINE( 'updated successfully' );
else
raise_application_error(-20001,'Error occurred while updating.');
end if;
end;
```

- **After deleted patient**

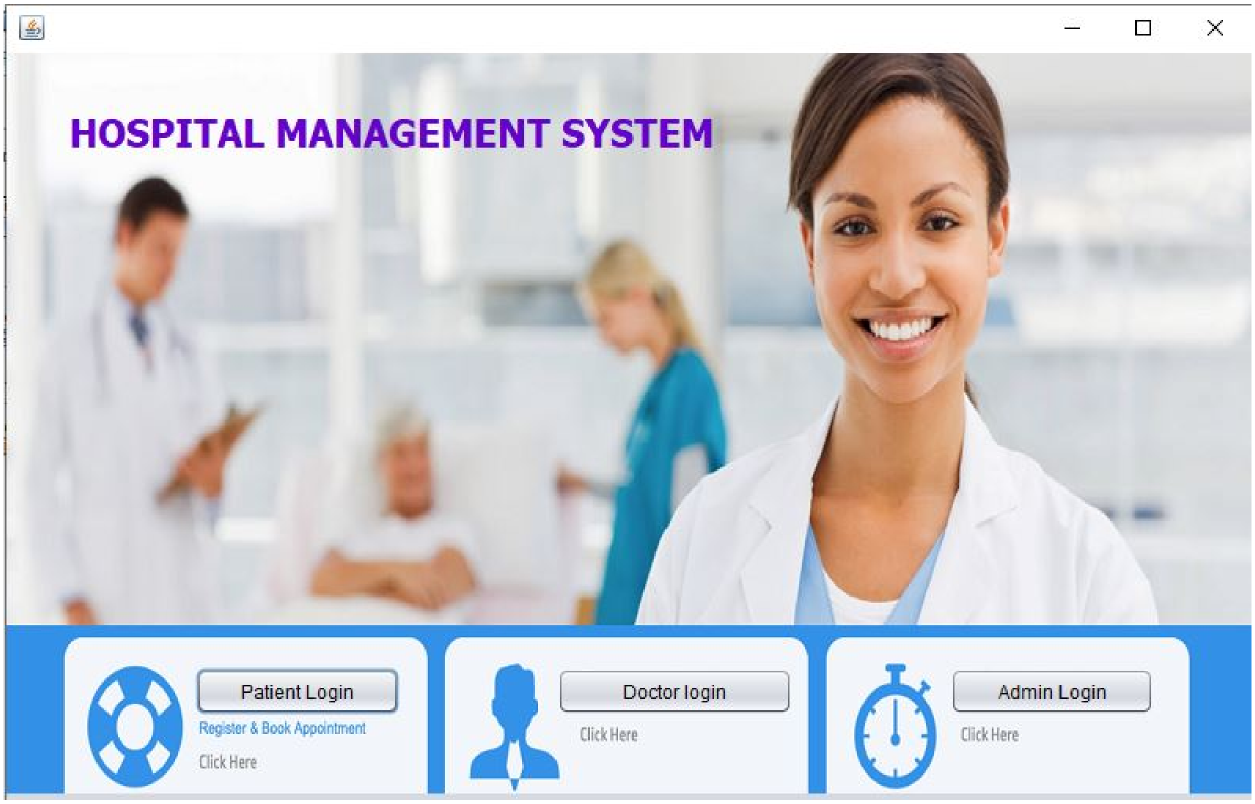
```
create or replace trigger tr_delete after delete on patients
for each row
begin
```

```
if deleting then
  DBMS_OUTPUT.PUT_LINE( 'deleted successfully' );
else
raise_application_error(-20001,'Error occurred while deleting.');
end if;
end;
```

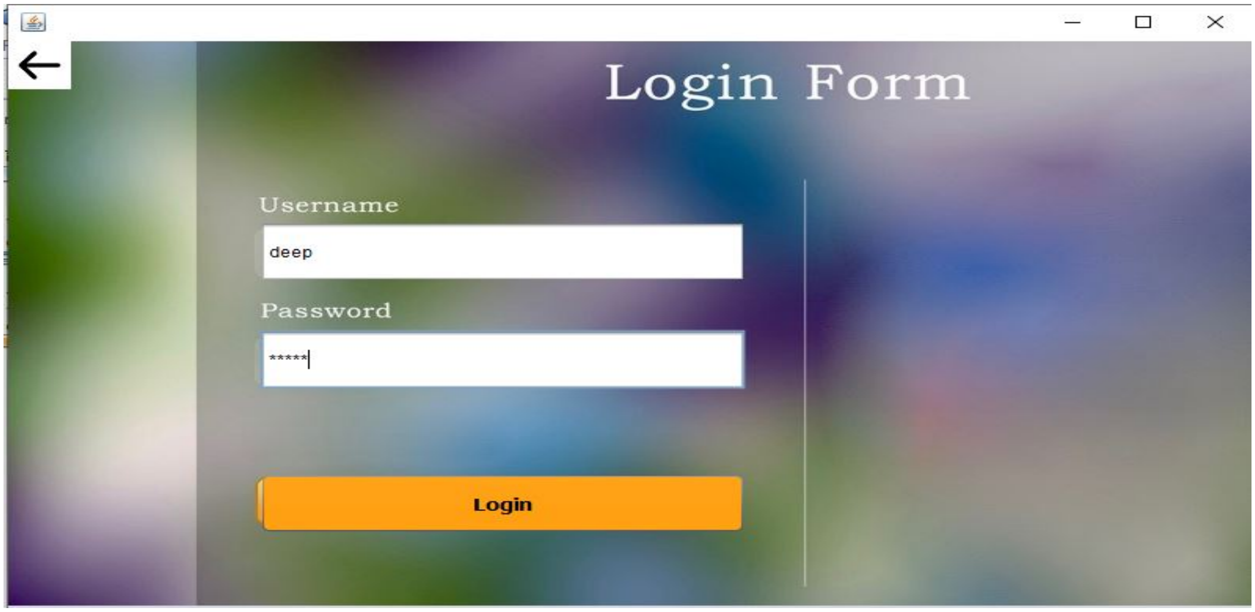
- **After inserting doctor**
- **After updating doctor**
- **After deleted doctor**

- **Screenshot Of Project**

- **Home Page**



- **Admin Login Page**



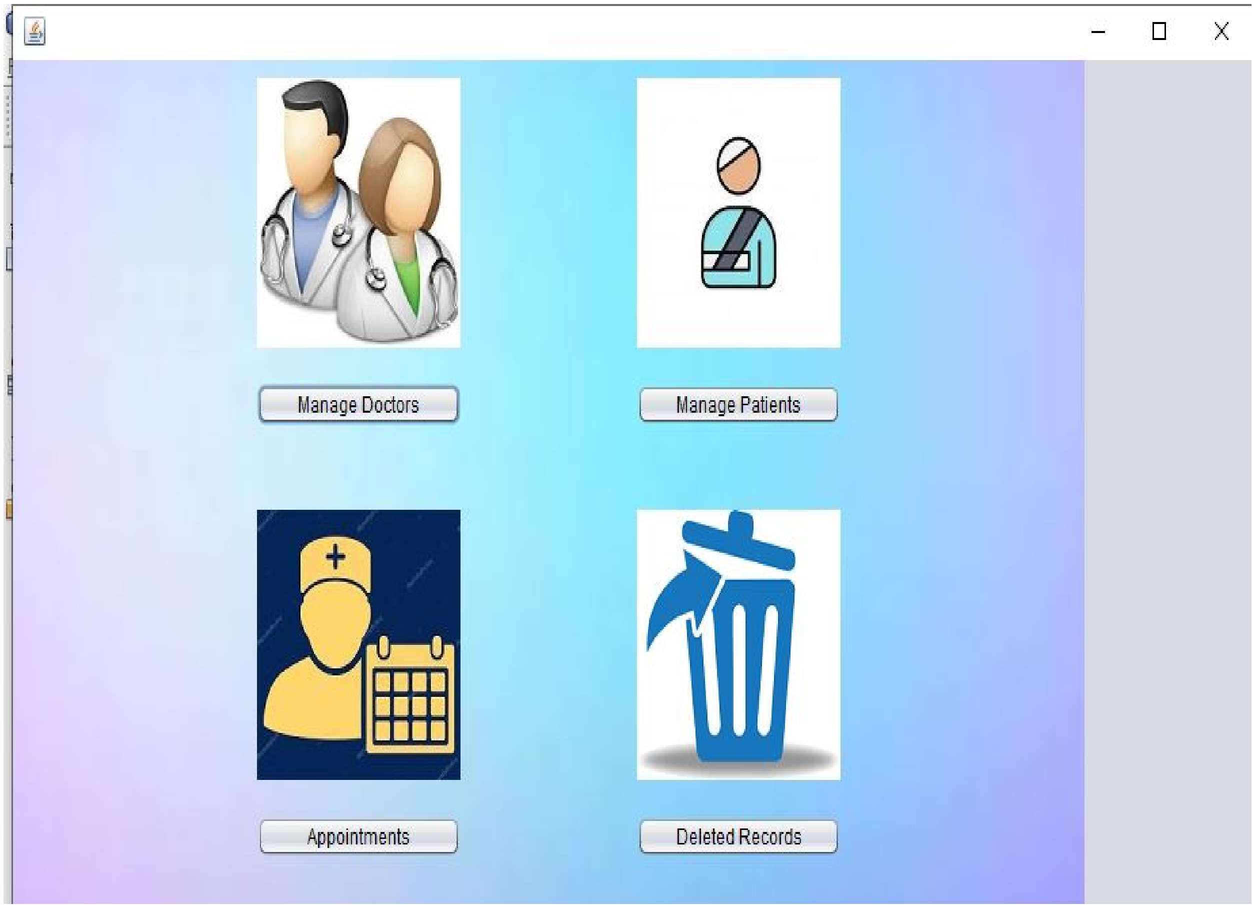
Username

deep

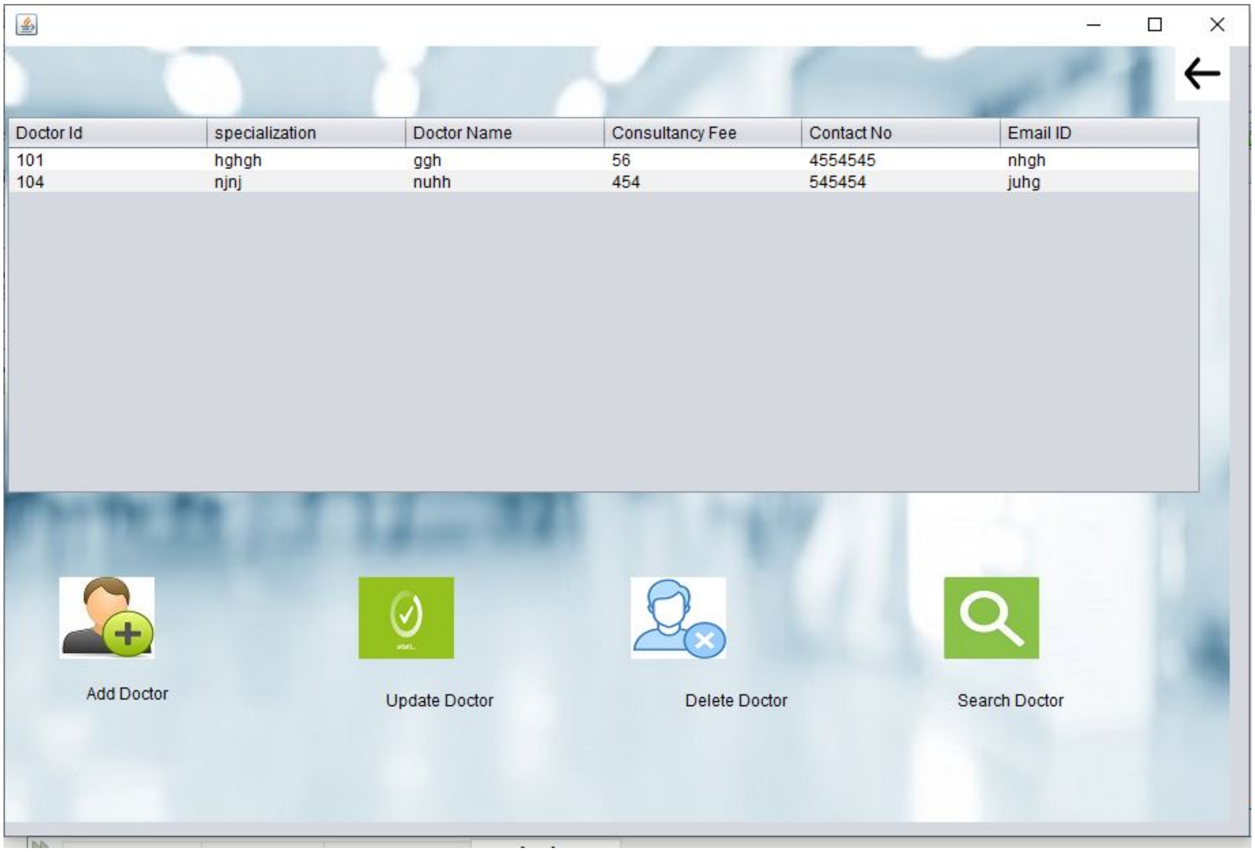
Password

*****

- **Admin Page**

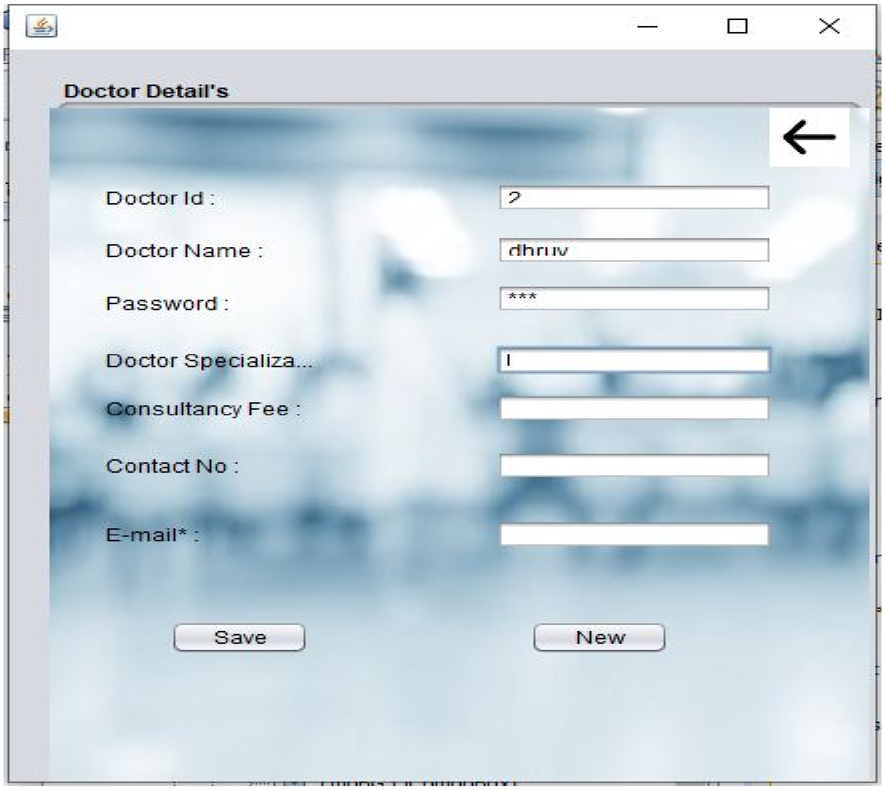


- **Manage Doctors Page**



Doctor Id	specialization	Doctor Name	Consultancy Fee	Contact No	Email ID
101	hghgh	ggh	56	4554545	nhgh
104	njnj	nuhh	454	545454	juhg

- **Add Doctor**



Doctor Detail's

Doctor Id : 2

Doctor Name : dhruv

Password : ***

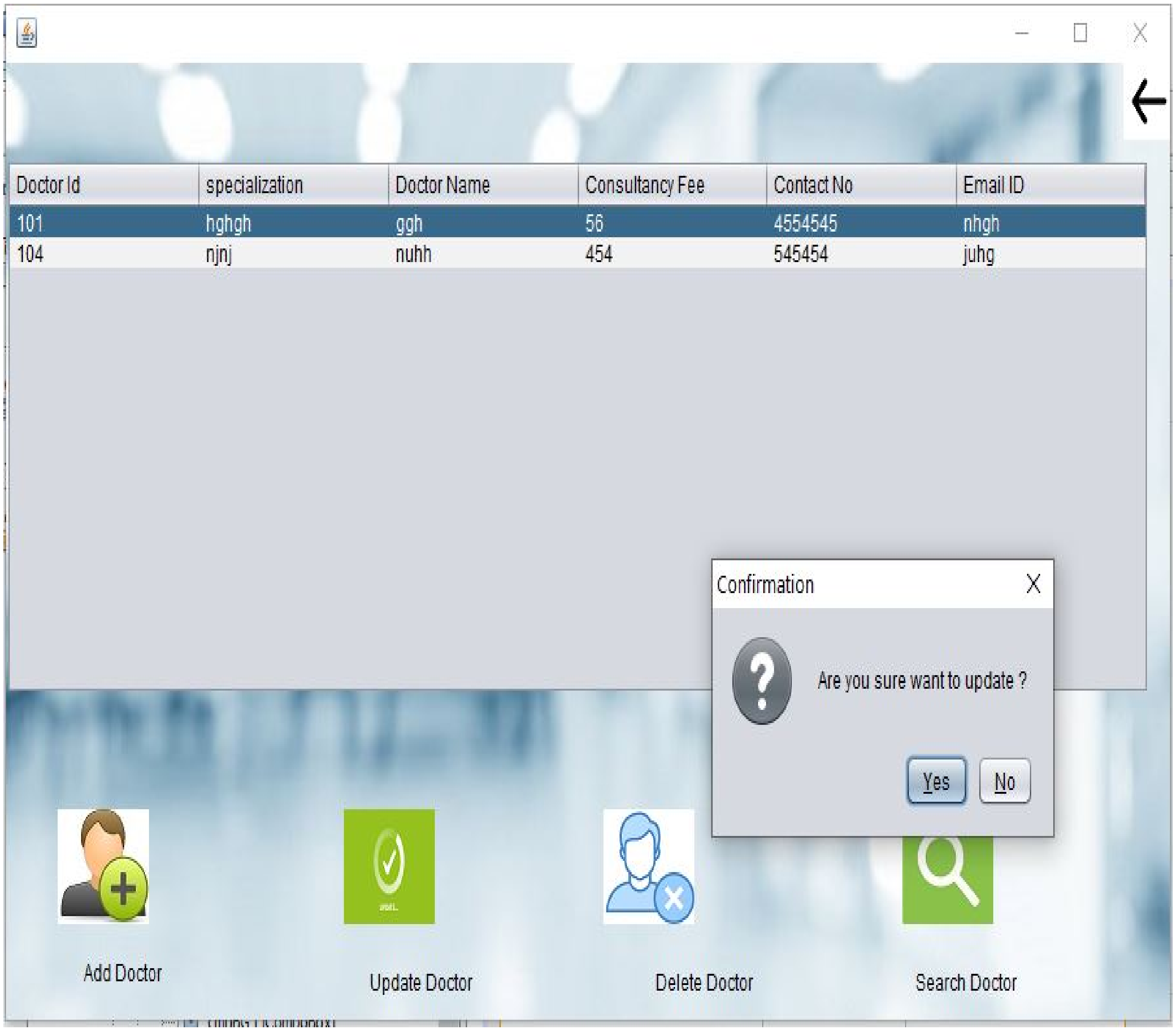
Doctor Specializa...

Consultancy Fee :

Contact No :

E-mail* :

- **Update Doctor**

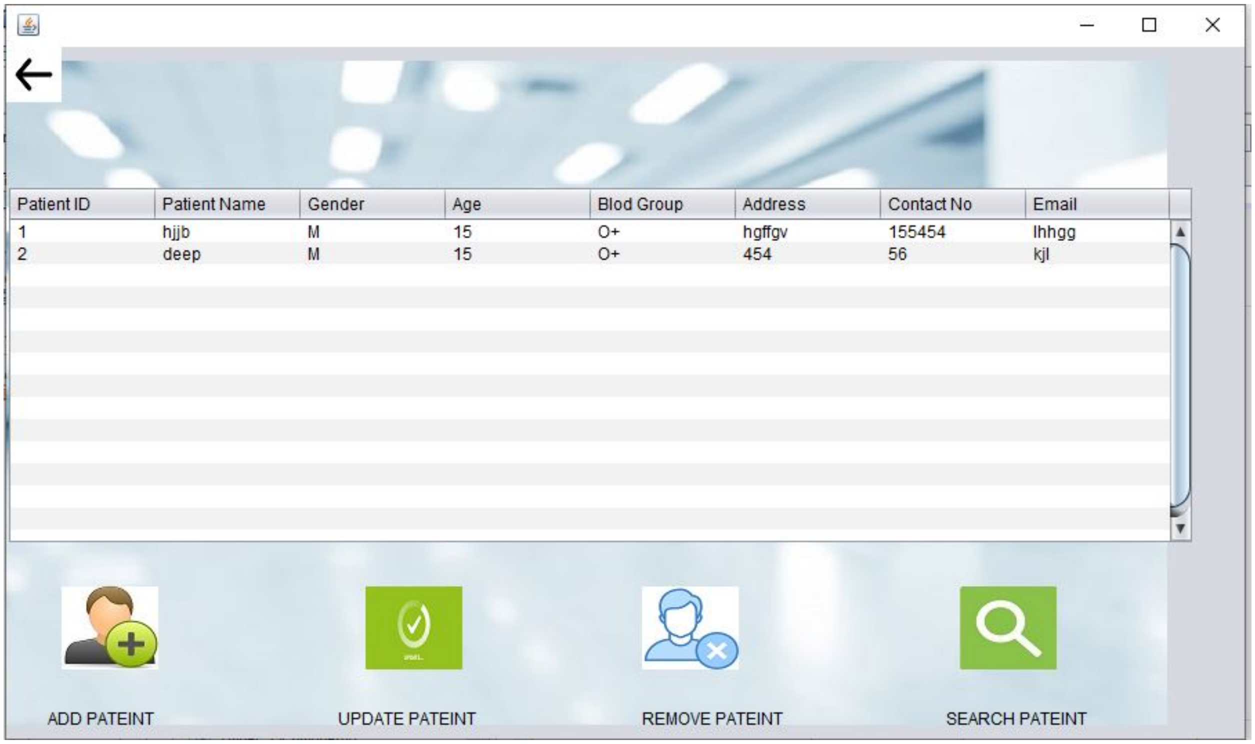


Doctor Id	specialization	Doctor Name	Consultancy Fee	Contact No	Email ID
101	hghgh	ggh	56	4554545	nhgh
104	njnj	nuhh	454	545454	juhg

Confirmation

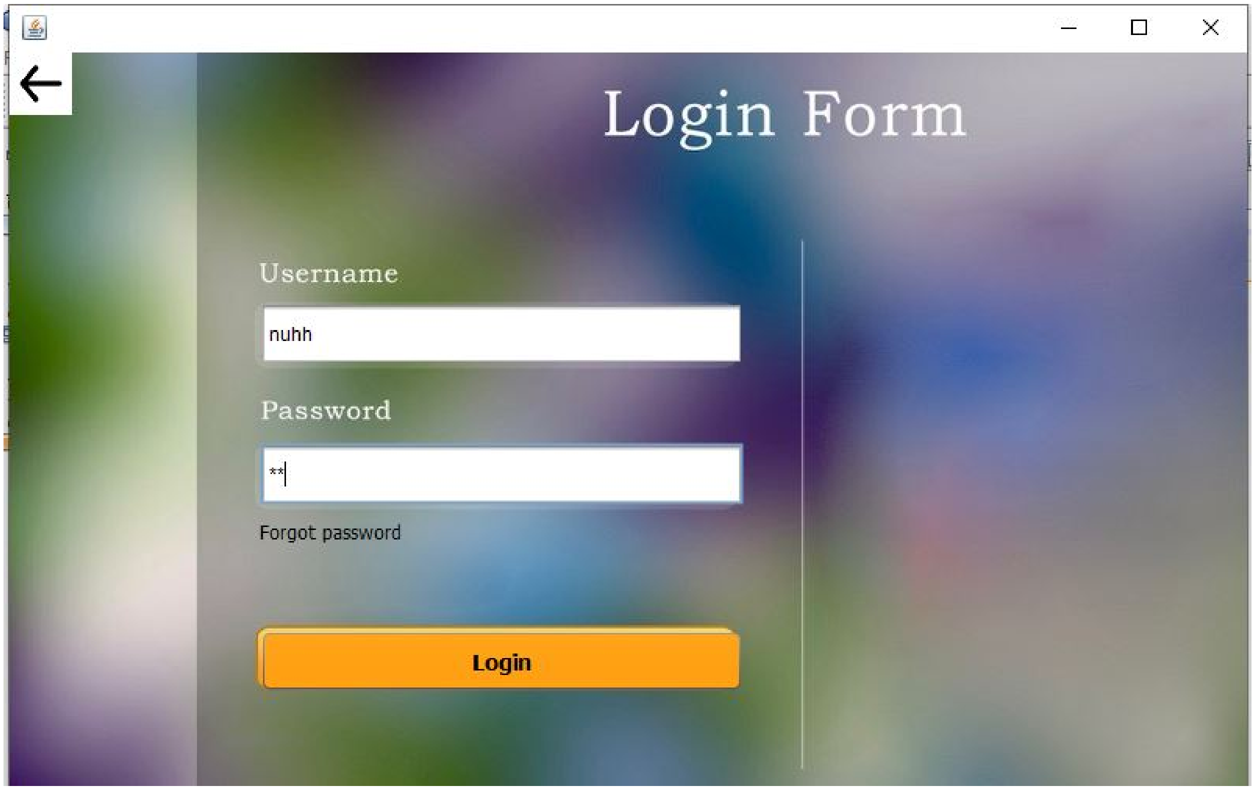
Are you sure want to update ?

- **Manage Patients**



Patient ID	Patient Name	Gender	Age	Blod Group	Address	Contact No	Email
1	hjjb	M	15	O+	hgffgv	155454	lhhgg
2	deep	M	15	O+	454	56	kjl

- **Doctor Login page**



Login Form

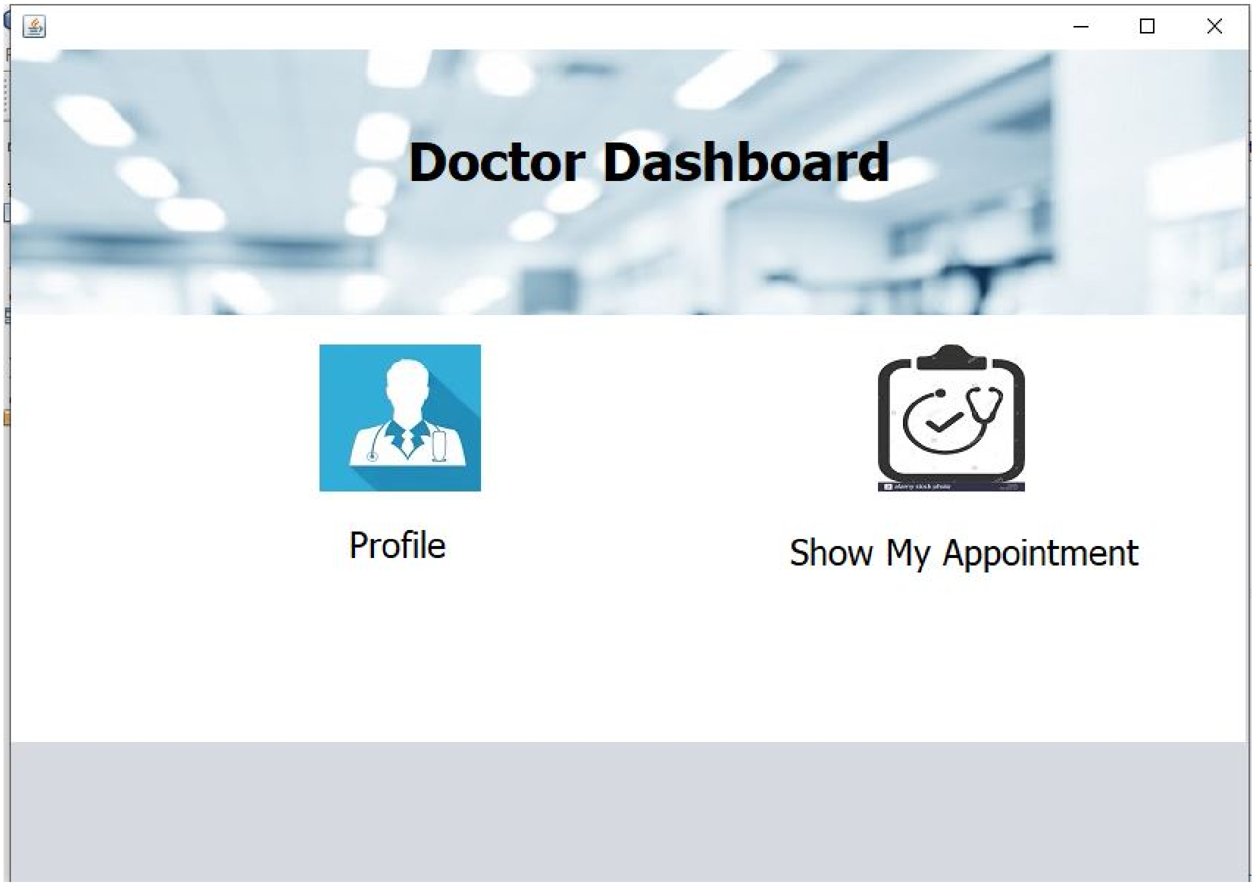
Username

nuhh

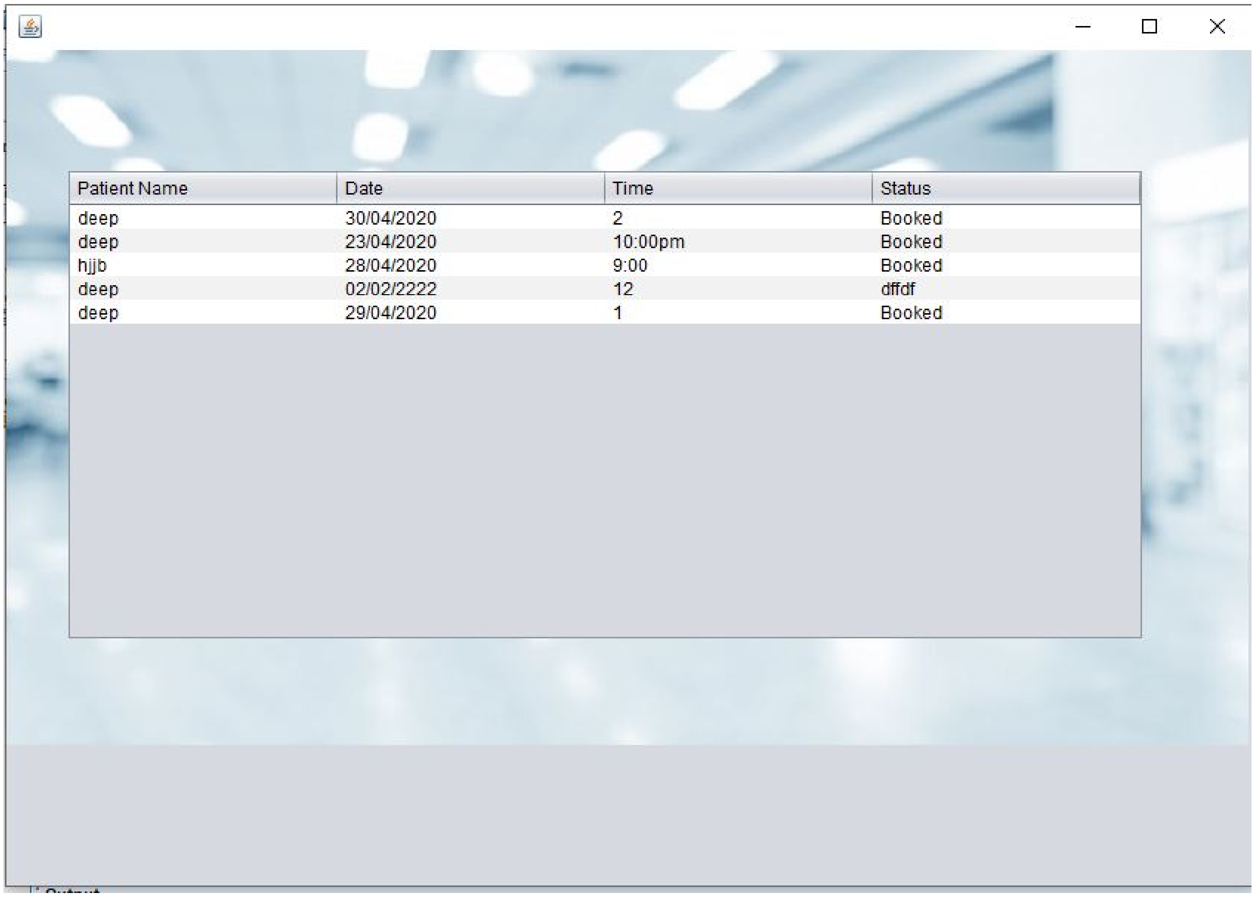
Password

**

- **Doctor Dashboard**

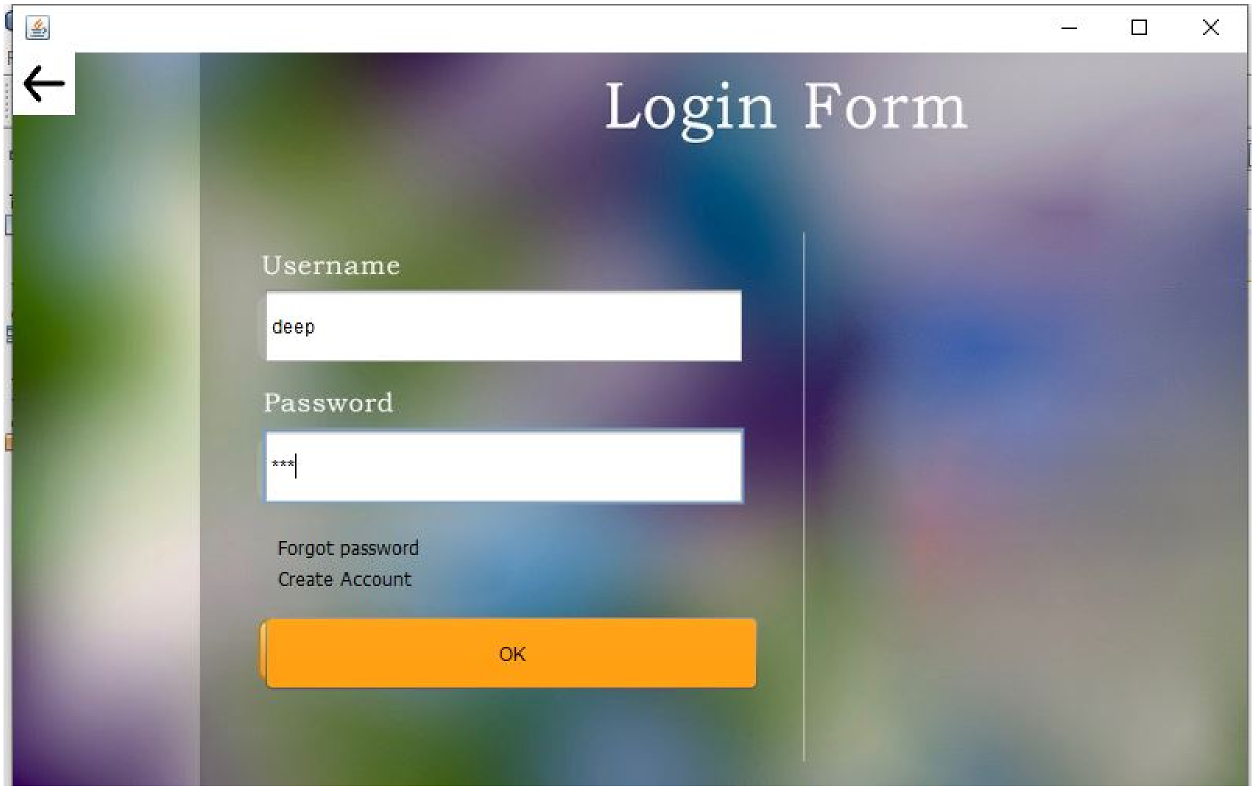


- **View Appointments**



Patient Name	Date	Time	Status
deep	30/04/2020	2	Booked
deep	23/04/2020	10:00pm	Booked
hjjb	28/04/2020	9:00	Booked
deep	02/02/2222	12	dffdf
deep	29/04/2020	1	Booked

- **Patients Login Page**



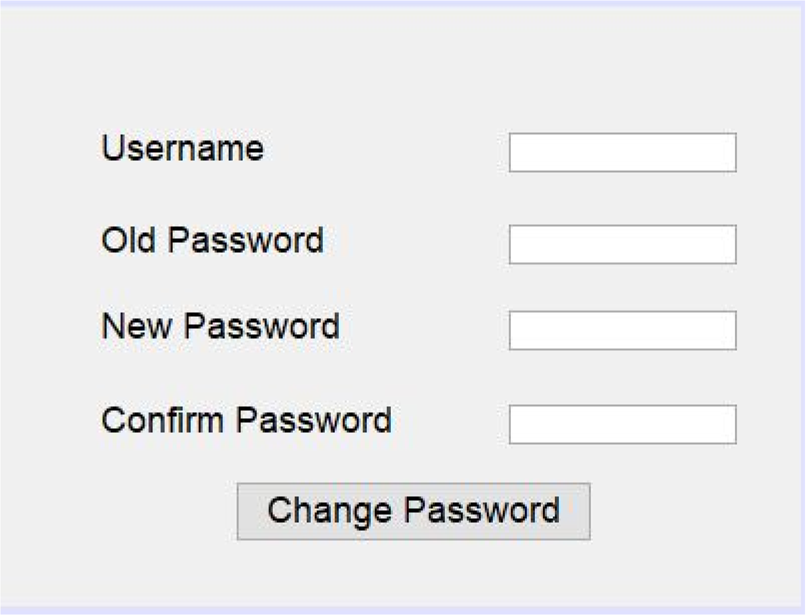
Username

deep

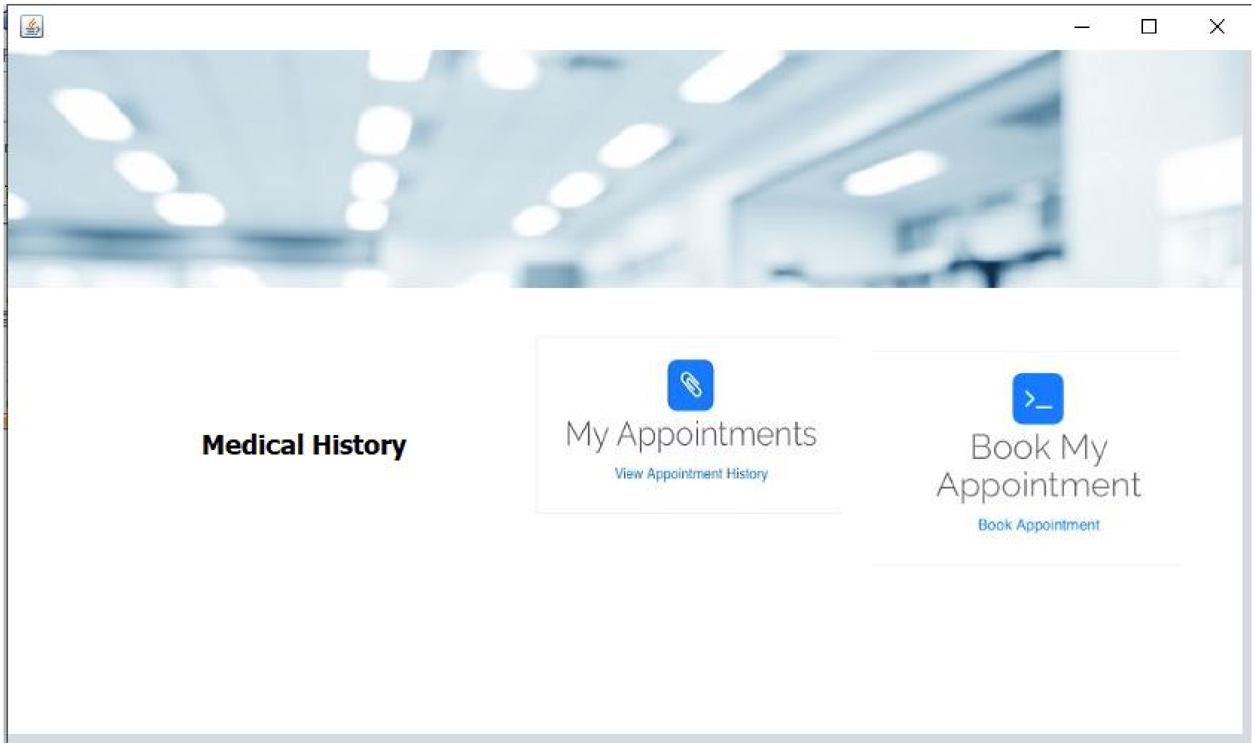
Password

***

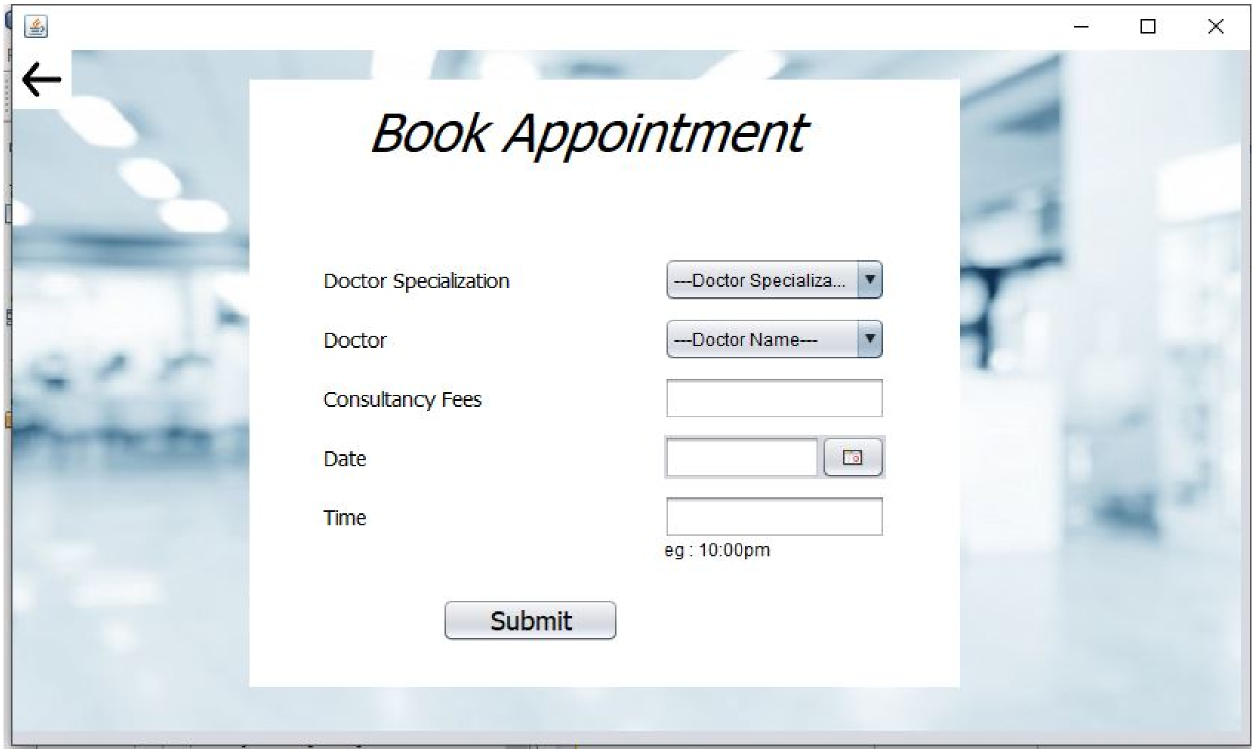
- **Forgot Password**



- **Patients Dashboard**



- **Book Appointments**



Book Appointment

Doctor Specialization ---Doctor Specializa...

Doctor ---Doctor Name---

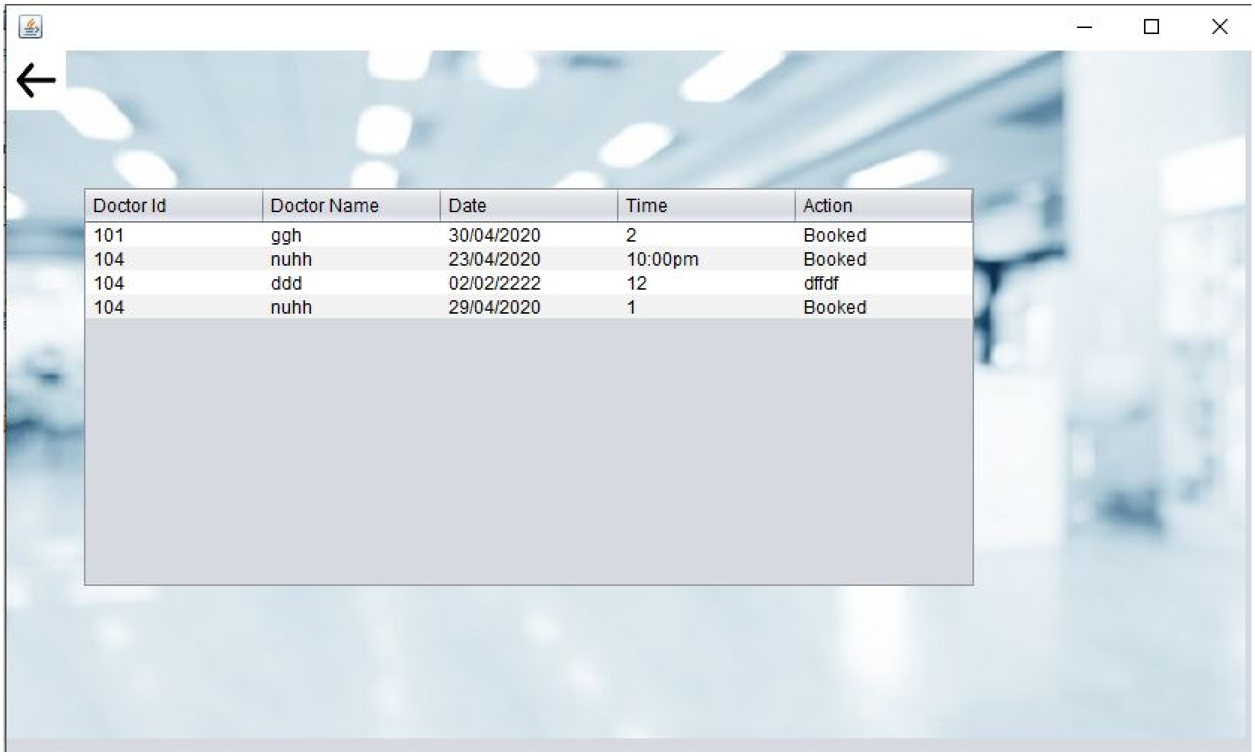
Consultancy Fees

Date

Time

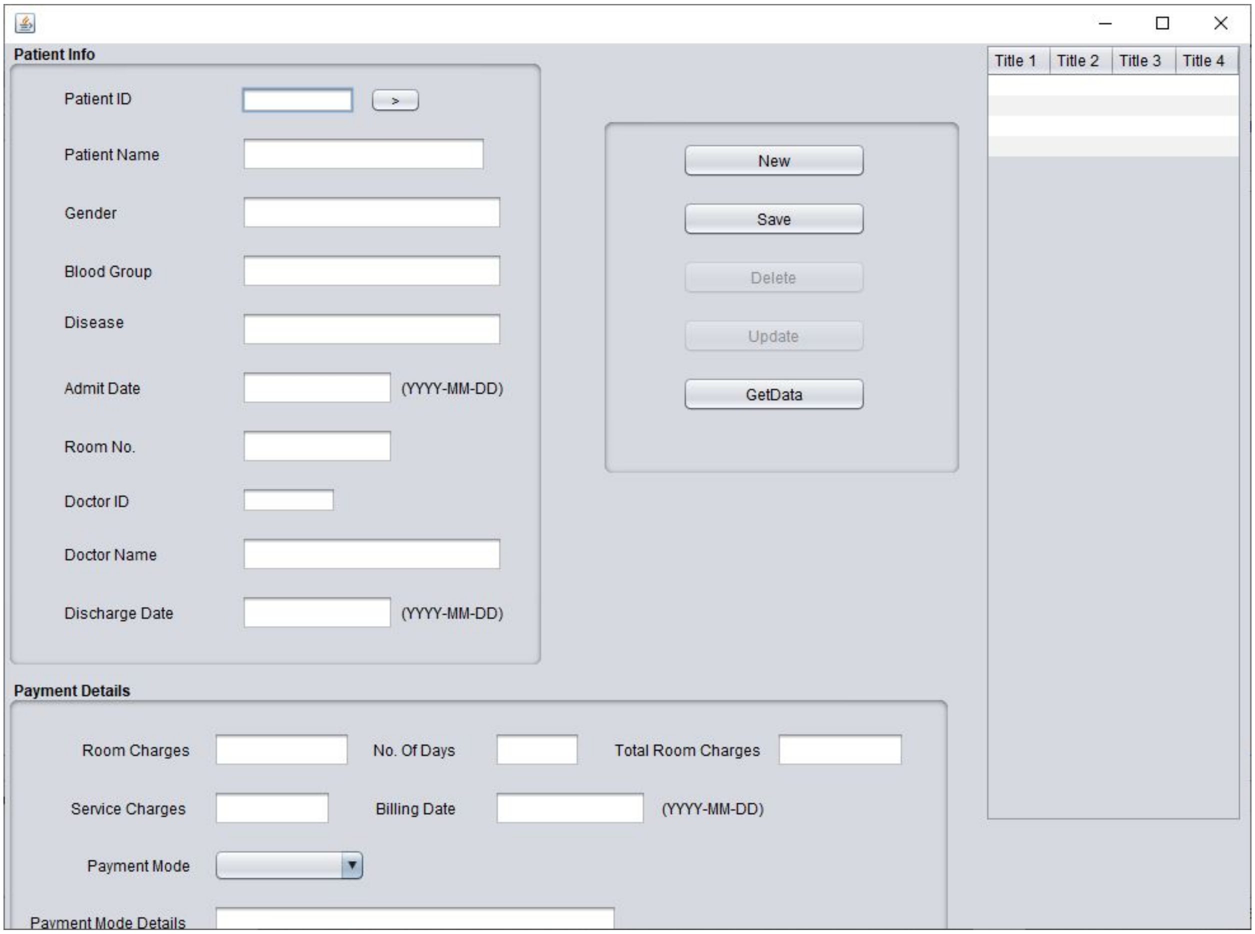
eg : 10:00pm

- **View Booked Appointments**

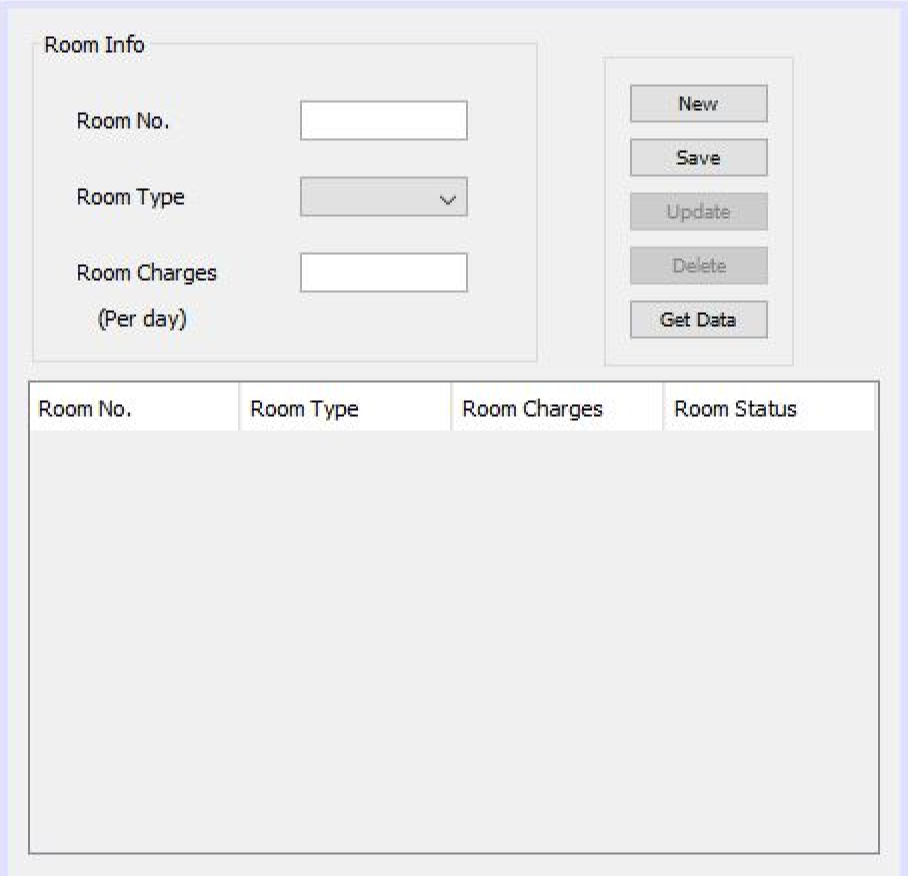


Doctor Id	Doctor Name	Date	Time	Action
101	ggh	30/04/2020	2	Booked
104	nuhh	23/04/2020	10:00pm	Booked
104	ddd	02/02/2222	12	dffdf
104	nuhh	29/04/2020	1	Booked

- **Generate Bill Page**

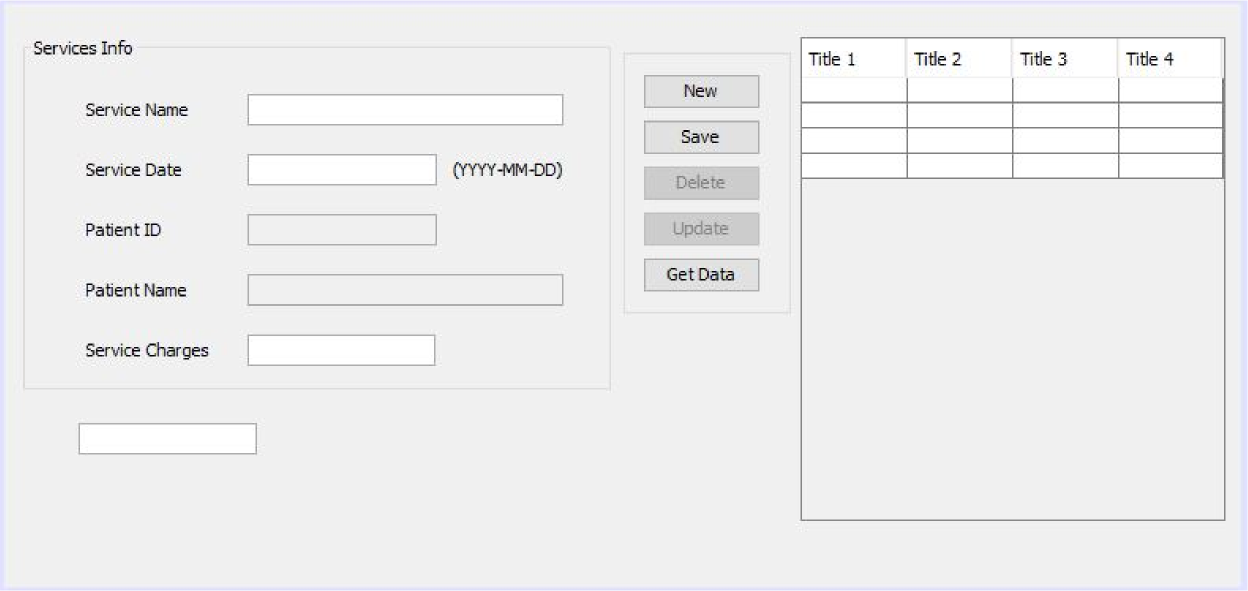


- **Room Information Page**



Room No.	Room Type	Room Charges	Room Status

- **Extra Services Page**



Title 1	Title 2	Title 3	Title 4