AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

Faculty of Engineering

Department of Computer Science & Engineering

Advanced Database Management

Section B

Hospital Management System

Midterm Report

Name	ID
Md. Nobobi Hasan	18-38712-3
Mufrad Mustavi	18-38776-3
Md. Abu Jubaer	18-38784-3
Md. Tanvir Shahriar	18-38787-3

Table of Contents

Summary:

In this project, we develop a management system called Hospital Management System. This system manages the hospital functionalities and events. The purpose of the Hospital Management System is to make a secure and easy way of storing information of the patient, doctors, staffs, laboratorists, accountants, inpatient, outpatient, rooms, and bill payment.

There are 6 different roles in this system:

Admin: An admin can register an admin, add/edit/delete staff, laboratorist, accountant and handle doctors, check their information, change password.

Staff: A staff handles patient, inpatient and outpatient information, decide room number for the patient.

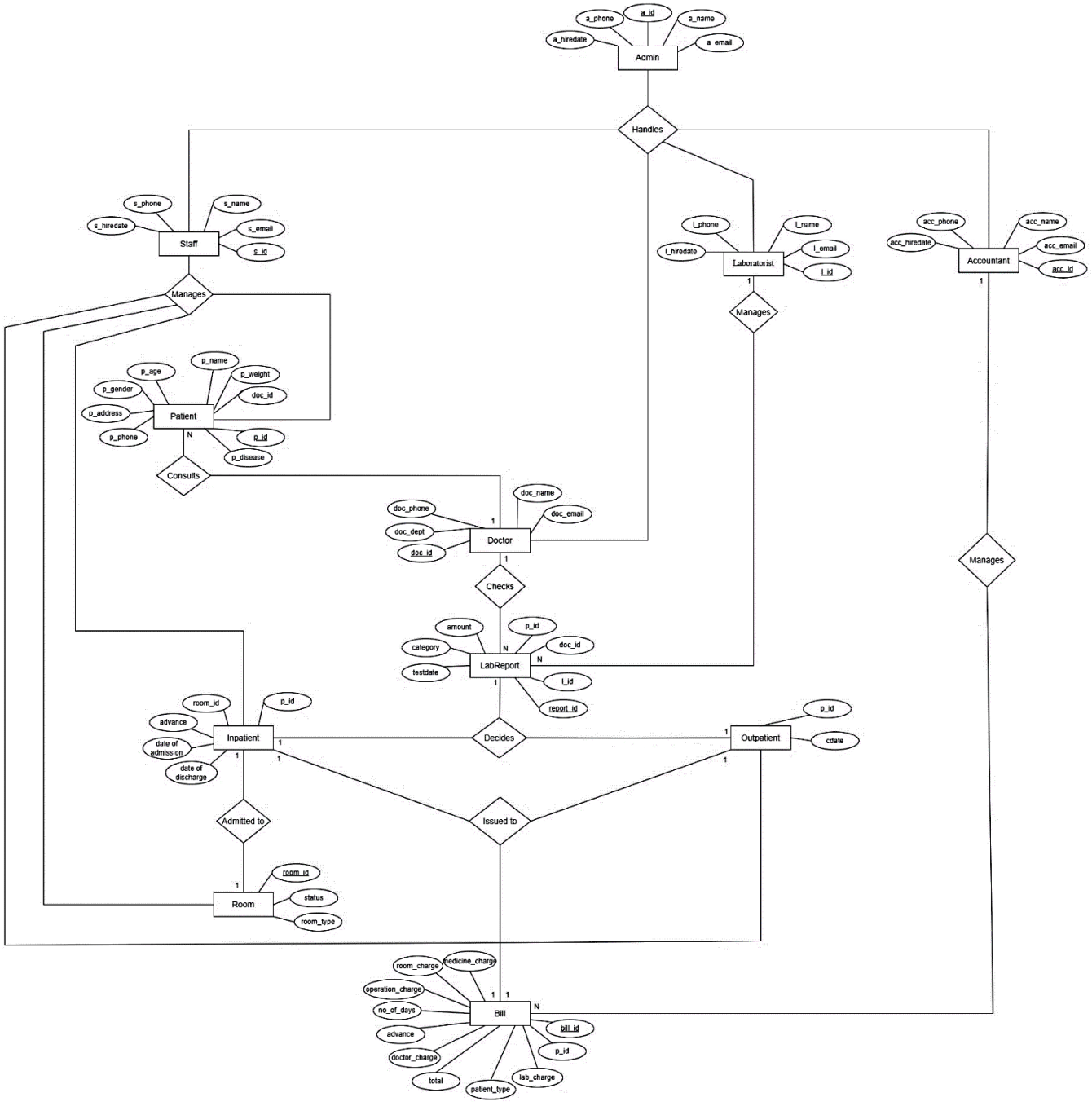
Laboratorist: A laboratorist handles lab report of a patient.

Accountant: An accountant handles all the bills and payments of a patient.

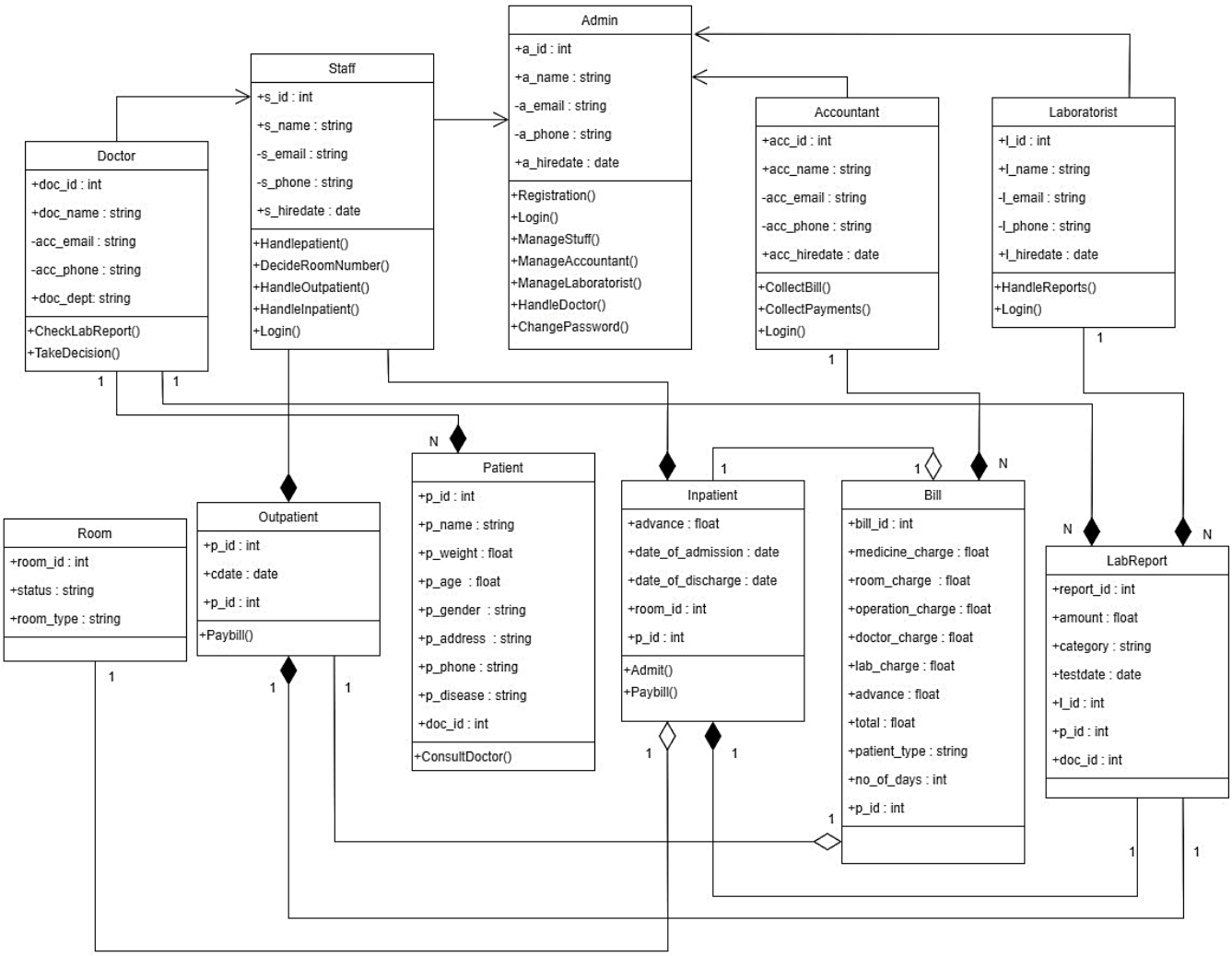
Doctor: A doctor checks the patient lab reports and decide which patient should stay in the hospital or not.

Patient: A patient consults with the doctor through staff.

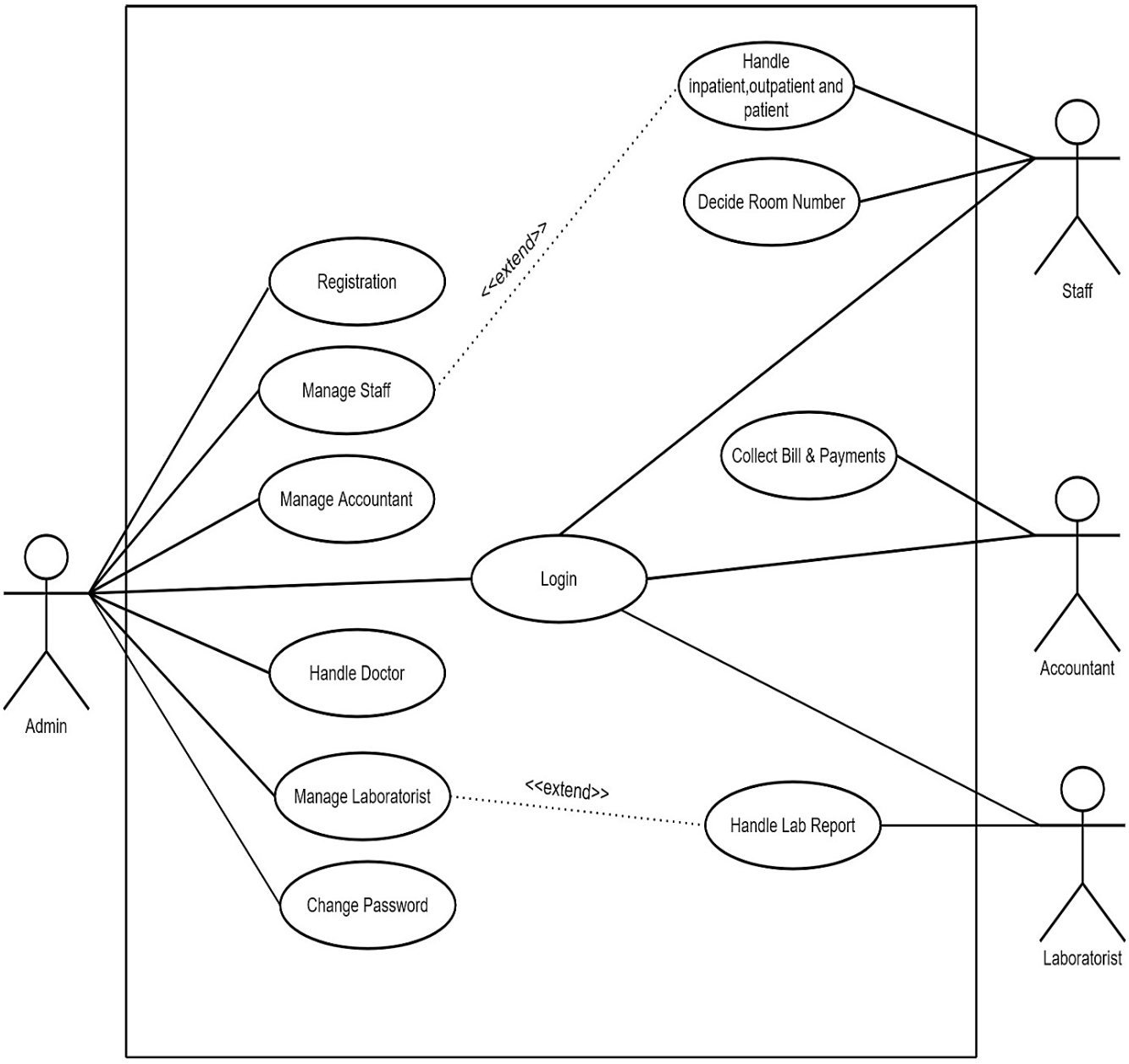
ER Diagram:


Admin
a_id
a_name
a_email
a_phone
a_hiredate
Handles
Staff
s_id
s_name
s_email
s_phone
s_hiredate
Manages
Laboratorist
l_id
l_name
l_email
l_phone
l_hiredate
Manages
Accountant
acc_id
acc_name
acc_email
acc_phone
acc_hiredate
Manages
Patient
p_id
p_name
p_age
p_gender
p_address
p_phone
p_weight
doc_id
p_disease
Consults
Doctor
doc_id
doc_name
doc_email
doc_phone
doc_dept
Checks
LabReport
report_id
amount
category
testdate
p_id
doc_id
l_id
Decides
Inpatient
p_id
room_id
advance
date of admission
date of discharge
Outpatient
p_id
cdate
Admitted to
Issued to
Room
room_id
status
room_type
Bill
bill_id
p_id
room_charge
medicine_charge
operation_charge
no_of_days
advance
doctor_charge
total
patient_type
lab_charge

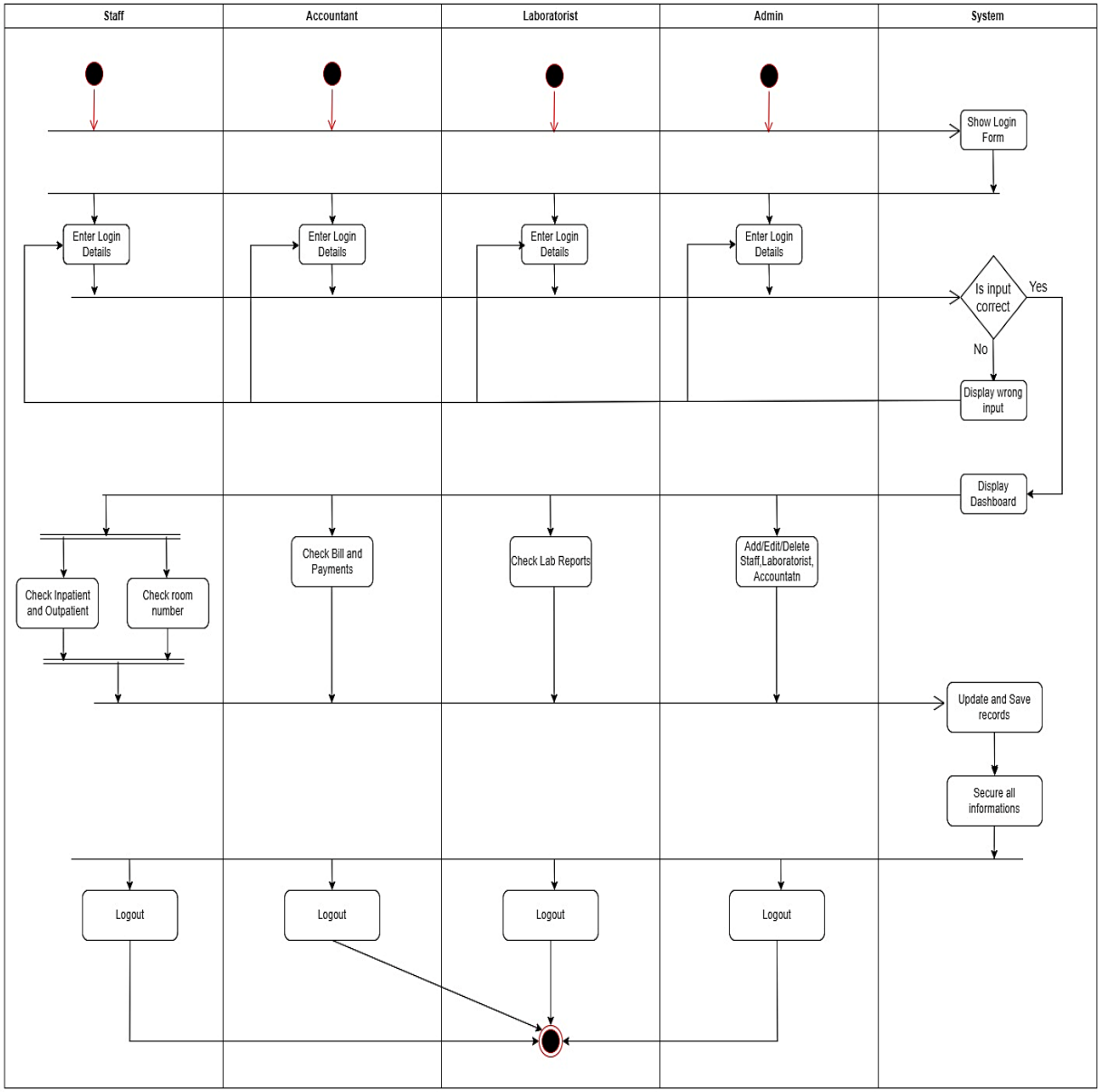
Class Diagram:



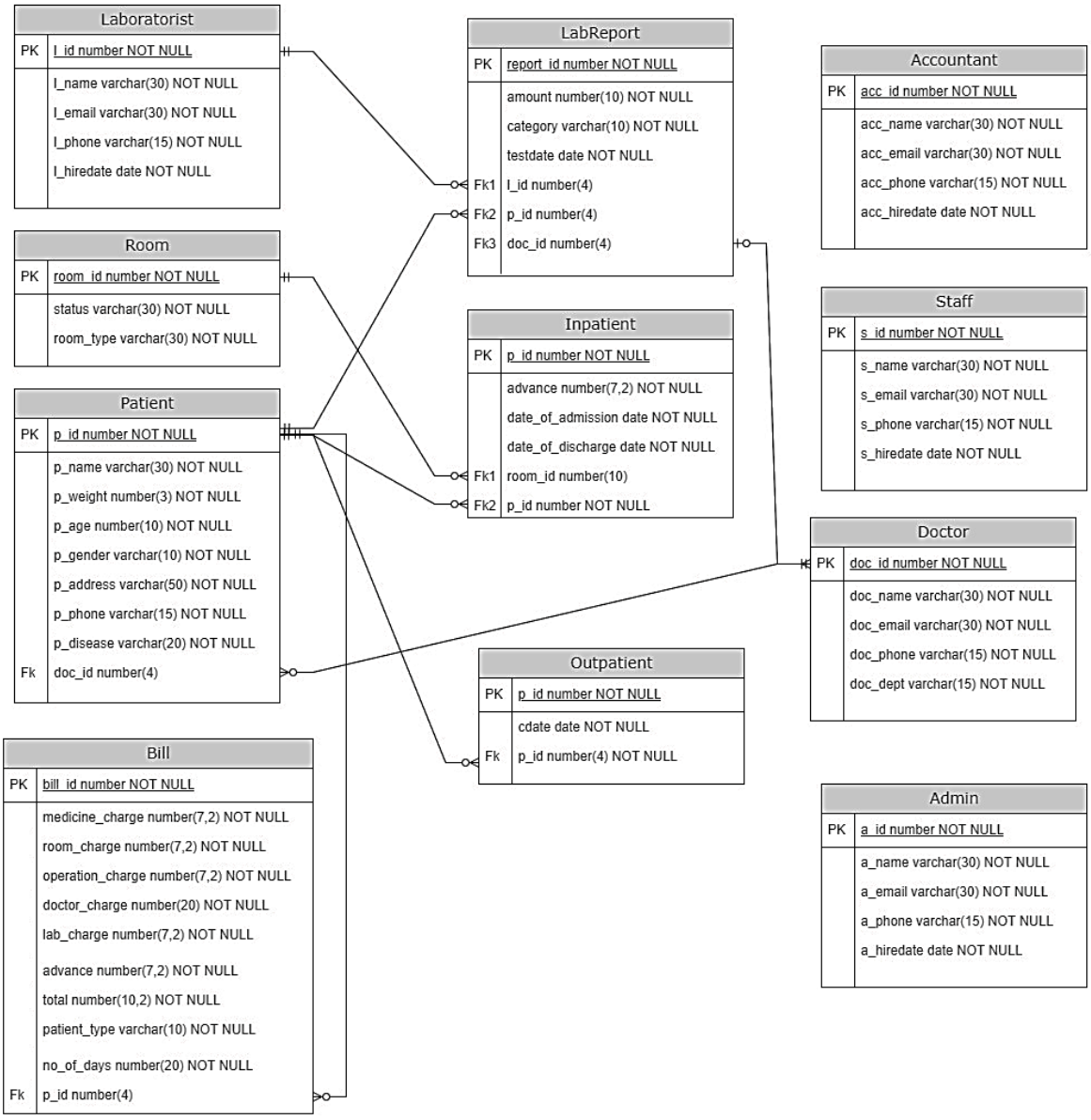
Use case Diagram:



Activity Diagram:



Schema diagram:



Tables:

Results Explain Describe Saved SQL History

A_ID	A_NAME	A_EMAIL	A_PHONE	A_HIREDATE	A_PASSWORD
1	admin1	a1@gmail.com	123456	02/22/1981	a111
2	admin2	a2@gmail.com	685498	02/22/2021	a222
3	admin3	a3@gmail.com	489985	02/22/2022	a333
4	admin4	a4@gmail.com	676765	02/22/2020	a444
5	admin5	a5@gmail.com	567868	02/22/2019	a555

5 rows returned in 0.00 seconds Download

Admins Table

Results Explain Describe Saved SQL History

ACC_ID	ACC_NAME	ACC_EMAIL	ACC_PHONE	ACC_HIREDATE	ACC_PASSWORD
1	accountant1	acc1@gmail.com	123456	03/22/1997	acc111
2	accountant2	acc2@gmail.com	685498	03/22/2021	acc222
3	accountant3	acc3@gmail.com	489985	03/22/2022	acc333
4	accountant4	acc4@gmail.com	676765	03/22/2020	acc444
5	accountant5	acc5@gmail.com	567868	03/22/2019	acc555

5 rows returned in 0.01 seconds Download

Accountants Table

Results Explain Describe Saved SQL History

S_ID	S_NAME	S_EMAIL	S_PHONE	S_HIREDATE	S_PASSWORD
1	staff1	staff1@gmail.com	543543	04/22/1997	s111
2	staff2	staff2@gmail.com	453145	04/22/2021	s222
3	staff3	staff3@gmail.com	354868	04/22/2022	s333
4	staff4	staff4@gmail.com	468468	04/22/2020	s444
5	staff5	staff5@gmail.com	534883	04/22/2019	s555

5 rows returned in 0.00 seconds Download

Staffs Table

Results Explain Describe Saved SQL History

L_ID	L_NAME	L_EMAIL	L_PHONE	L_HIREDATE	L_PASSWORD
1	laboratorist1	laboratorist1@gmail.com	543543	05/22/1997	l111
2	laboratorist2	laboratorist2@gmail.com	453145	05/22/2021	l222
3	laboratorist3	laboratorist3@gmail.com	354868	05/22/2018	l333
4	laboratorist4	laboratorist4@gmail.com	468468	05/22/2020	l444
5	laboratorist5	laboratorist5@gmail.com	534883	05/22/2019	l555

5 rows returned in 0.00 seconds Download

Laboratorists Table

Results Explain Describe Saved SQL History

DOC_ID	DOC_NAME	DOC_EMAIL	DOC_PHONE	DOC_DEPT
1	doctor1	doctor1@gmail.com	456895	Cardiologists
2	doctor2	doctor2@gmail.com	498647	Dermatologists
3	doctor3	doctor3@gmail.com	548547	Cardiologists
4	doctor4	doctor4@gmail.com	254485	Neurologists
5	doctor5	doctor5@gmail.com	244856	Neurologists

5 rows returned in 0.01 seconds Download

Doctors Table

Results Explain Describe Saved SQL History

ROOM_ID	STATUS	ROOM_TYPE
1	not available	VIP
2	not available	VIP
3	available	VIP
4	not available	general
5	not available	general
6	not available	general
7	available	general

7 rows returned in 0.00 seconds Download

Rooms Table

Results Explain Describe Saved SQL History

P_ID	P_NAME	P_WEIGHT	P_AGE	P_GENDER	P_ADDRESS	P_PHONE	P_DISEASE	DOC_ID
1	patient1	56	24	male	wirelesspara	547893	chest pain	3
2	patient2	67	62	male	uttara	547893	headache	2
3	patient3	42	17	female	mohammadpur	547893	leg broken	1
4	patient4	75	32	female	basundhara	547893	arm broken	5
5	patient5	89	54	male	gulshan	547893	headache	4
6	patient6	56	24	male	wirelesspara	547893	chest pain	3
7	patient7	67	62	male	uttara	547893	headache	1
8	patient8	42	17	female	mohammadpur	547893	leg broken	2
9	patient9	75	32	female	basundhara	547893	stomach ache	1
10	patient10	89	54	male	gulshan	547893	headache	5

10 rows returned in 0.00 seconds Download

Patients Table

Results Explain Describe Saved SQL History

P_ID	ADVANCE	DATE_OF_ADMISSION	DATE_OF_DISCHARGE	ROOM_ID
1	500	06/17/2020	06/25/2020	1
3	500	04/22/2019	04/29/2019	2
4	500	02/09/2021	02/16/2021	3
6	500	08/05/2020	08/06/2020	4
8	500	09/25/2018	10/07/2018	5

5 rows returned in 0.00 seconds Download

Inpatients Table

Results Explain Describe Saved SQL History

P_ID	CDATE
2	06/17/2020
5	06/17/2021
7	06/17/2018
9	06/17/2019
10	06/17/2017

5 rows returned in 0.00 seconds Download

Outpatients Table

Results Explain Describe Saved SQL History

REPORT_ID	AMOUNT	CATEGORY	TESTDATE	L_ID	P_ID	DOC_ID
1	200	x-ray	06/17/2020	1	1	1
2	200	blood test	06/17/2019	2	2	2
3	200	blood test	06/18/2020	3	3	3
4	500	urine test	04/05/2018	4	4	4
5	200	x-ray	06/17/2020	5	5	5

5 rows returned in 0.00 seconds Download

Labreports Table

Results Explain Describe Saved SQL History

BILL_ID	MEDICINE_CHARGE	ROOM_CHARGE	OPERATION_CHARGE	DOCTOR_CHARGE	LAB_CHARGE	ADVANCE	TOTAL	PATIENT_TYPE	NO_OF_DAYS	P_ID
1	1500	9000	80000	25000	500	5000	115.5	inpatient	3	1
2	1500	9000.5	80000	25000	500	5000	115000.5	outpatient	0	2
3	1500	9000.5	80000	25000	500	5000	115000.5	inpatient	3	3
4	1500	9000.5	80000	25000	500	5000	115000.5	inpatient	3	4
5	1500	9000.5	80000	25000	500	5000	115000.5	outpatient	0	5

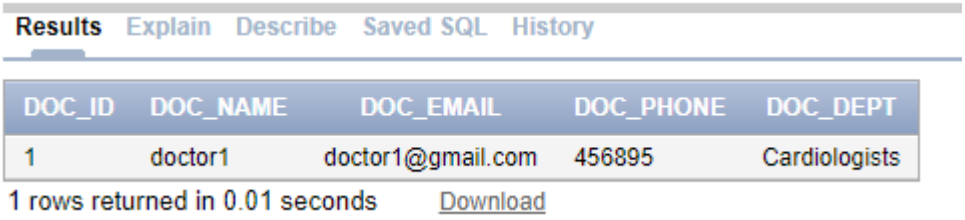
5 rows returned in 0.00 seconds Download

Bills Table

SQL Question with Queries:

1. **Find out the details of the doctor who's associated with most number of patients**

   => select * from doctor where doc_id = (select doc_id from(select doc_id, count(doc_id) as p_handel from patient group by doc_id) where p_handel = (select max(p_handel) from (select doc_id, count(doc_id) as p_handel from patient group by doc_id)))



Results Explain Describe Saved SQL History

DOC_ID	DOC_NAME	DOC_EMAIL	DOC_PHONE	DOC_DEPT
1	doctor1	doctor1@gmail.com	456895	Cardiologists

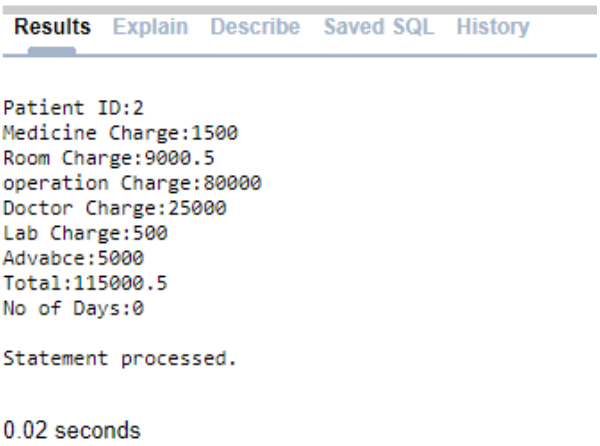
1 rows returned in 0.01 seconds Download

2. **Write a program that takes patient id as input and shows his/her billing details.**

   =>

```
declare
        pid bill.p_id%type:=:Enter_patient_id;
        mc bill.medicine_charge%type;
        rc bill.room_charge%type;
        oc bill.operation_charge%type;
        dc bill.doctor_charge%type;
        lc bill.lab_charge%type;
        adv bill.advance%type;
        tot bill.total%type;
        nod bill.no_of_days% type;

begin
        select medicine_charge, room_charge, operation_charge, doctor_charge,
lab_charge, advance, total, no_of_days into mc, rc, oc, dc, lc, adv, tot, nod from    bill
where p_id= pid;

        DBMS_OUTPUT.PUT_LINE('Patient ID:'||pid);
        DBMS_OUTPUT.PUT_LINE('Medicine Charge:'||mc);
        DBMS_OUTPUT.PUT_LINE('Room Charge:'||rc);
        DBMS_OUTPUT.PUT_LINE('operation Charge:'||oc);
        DBMS_OUTPUT.PUT_LINE('Doctor Charge:'||dc);
        DBMS_OUTPUT.PUT_LINE('Lab Charge:'||lc);
        DBMS_OUTPUT.PUT_LINE('Advabce:'||adv);
```

```
        DBMS_OUTPUT.PUT_LINE('Total:'||tot);
        DBMS_OUTPUT.PUT_LINE('No of Days:'||nod);

end;
```



3. **Create a program which will take the p_id of an patient and will return the updated total after giving discount. The discount will take place based on the total amount.**

> 100000 tk	5%
60000 to 100000 tk	4%
<60000 tk	3%

=>

```
declare
        pid bill.p_id%type:=:Enter_patient_id;
        tot bill.total%type;

begin
        select total into tot from bill where p_id=pid;

        if(tot > 100000) then
        update bill set total = total - (total*0.05) where p_id=pid;
        DBMS_OUTPUT.PUT_LINE('Patient ID: ' || pid);
        DBMS_OUTPUT.PUT_LINE('New total: ' || tot);

        elsif(tot > 60000 and tot <100000) then
        update bill set total = total - (total*0.04) where p_id=pid;
        DBMS_OUTPUT.PUT_LINE('Patient ID: ' || pid);
        DBMS_OUTPUT.PUT_LINE('New total: ' || tot);

        elsif(tot < 60000) then
```

```
        update bill set total = total - (total*0.03) where p_id=pid;
        DBMS_OUTPUT.PUT_LINE('Patient ID: ' || pid);
        DBMS_OUTPUT.PUT_LINE('New total: ' || tot);


    else
        DBMS_OUTPUT.PUT_LINE('Can not be updated!');

    end if;

    end;
```

Results Explain Describe Saved SQL History

```
Patient ID: 2
New total: 115000.5

1 row(s) updated.

0.00 seconds
```

4. **Find out the details of oldest admin.**

=>select * from admin where a_hiredate = (select min(a_hiredate) from admin);

Results Explain Describe Saved SQL History

A_ID	A_NAME	A_EMAIL	A_PHONE	A_HIREDATE	A_PASSWORD
1	admin1	a1@gmail.com	123456	02/22/1981	a111

1 rows returned in 0.00 seconds Download

5. **Find out the patient whose associated doctor is same as patient3.**

=> select f.* from patient f, patient r where r.doc_id = f.doc_id and r.p_name = 'patient3' and f.p_name!='patient3';

Results Explain Describe Saved SQL History

P_ID	P_NAME	P_WEIGHT	P_AGE	P_GENDER	P_ADDRESS	P_PHONE	P_DISEASE	DOC_ID
7	patient7	67	62	male	uttara	547893	headache	1
9	patient9	75	32	female	basundhara	547893	stomach ache	1

2 rows returned in 0.00 seconds Download

6. **Find out the name of diseases who have done blood test.**

   => select p.p_disease, p.p_id from patient p, labreport r where p.p_id = r.p_id and category = 'blood test'

Results Explain Describe Saved SQL History

P_DISEASE	P_ID
headache	2
leg broken	3

2 rows returned in 0.00 seconds Download

7. **Find out the patients who have done x-ray and give them a 200 tk discount on their lab_charge.**

=> update bill set lab_charge = lab_charge - 200 where p_id in (select p_id from labreport where category = 'x-ray');

Results Explain Describe Saved SQL History

```
2 row(s) updated.
```

0.01 seconds

8. **Find out the average doctor charge of doctors.**
   => select avg(b.doctor_charge), p.doc_id from bill b, patient p where p.p_id=b.p_id group by p.doc_id

Results Explain Describe Saved SQL History

AVG(B.DOCTOR_CHARGE)	DOC_ID
25000	1
25000	2
25000	5
25000	4
25000	3

5 rows returned in 0.00 seconds Download

9. Create a program which will take the p_id of an patient and will return the patient type (inpatient or outpatient)

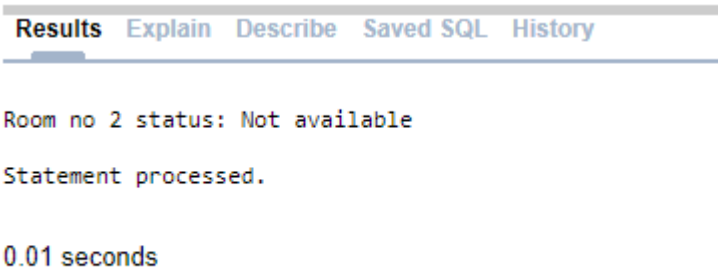
```
=>  declare
        pid patient.p_id%type:=:Enter_patient_id;
        inp number:= 0;
        outp number:= 0;

begin
        select count(*) into inp from inpatient where p_id=pid;
        select count(*) into outp from outpatient where p_id=pid;

if(inp > 0) then
        DBMS_OUTPUT.PUT_LINE('Patient type: InPatient');

elsif(outp > 0) then
        DBMS_OUTPUT.PUT_LINE('Patient type: OutPatient');

else
        DBMS_OUTPUT.PUT_LINE('Patient not found!');

end if;

end;
```

Results Explain Describe Saved SQL History

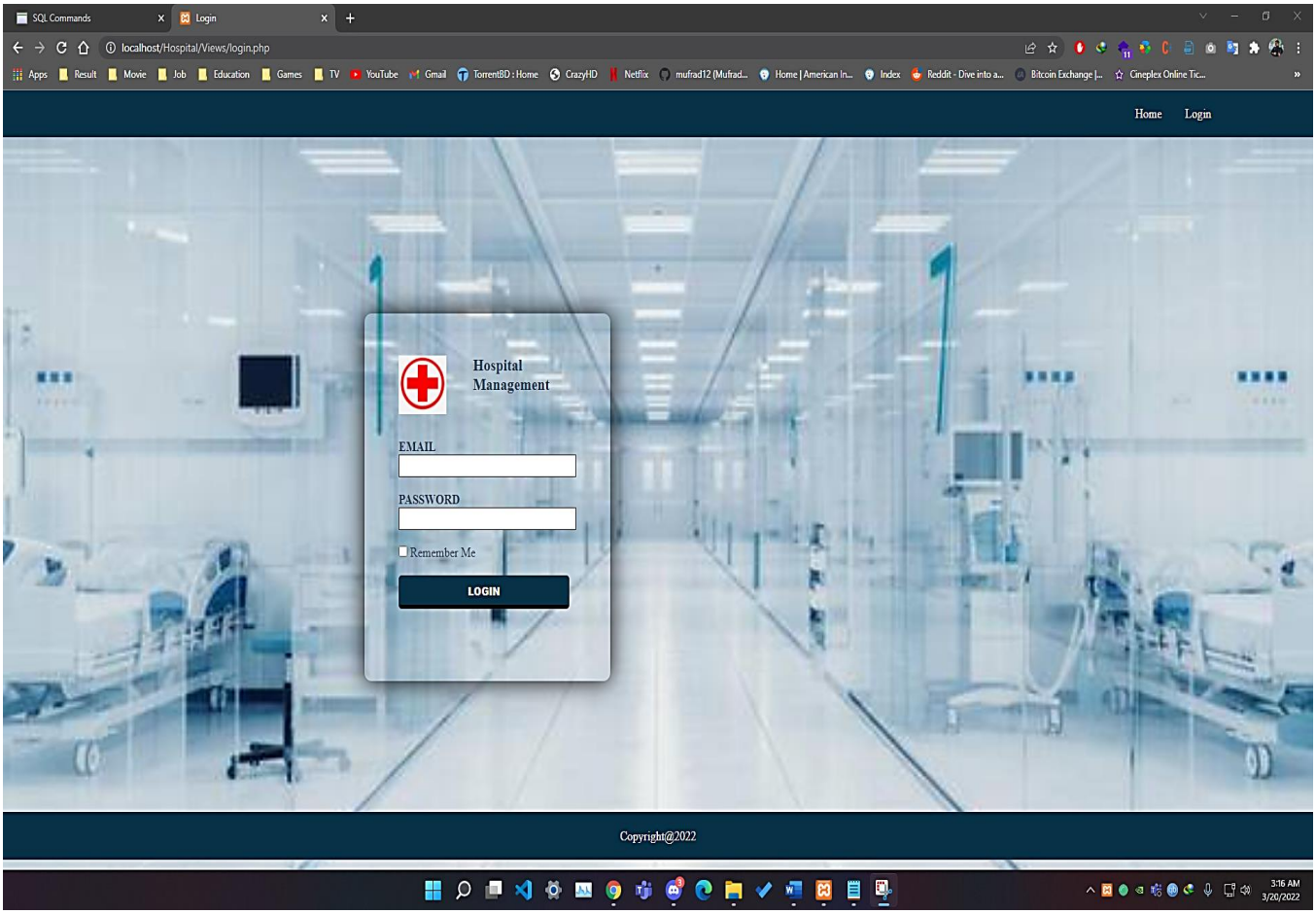
```
Patient type: OutPatient

Statement processed.

0.01 seconds
```

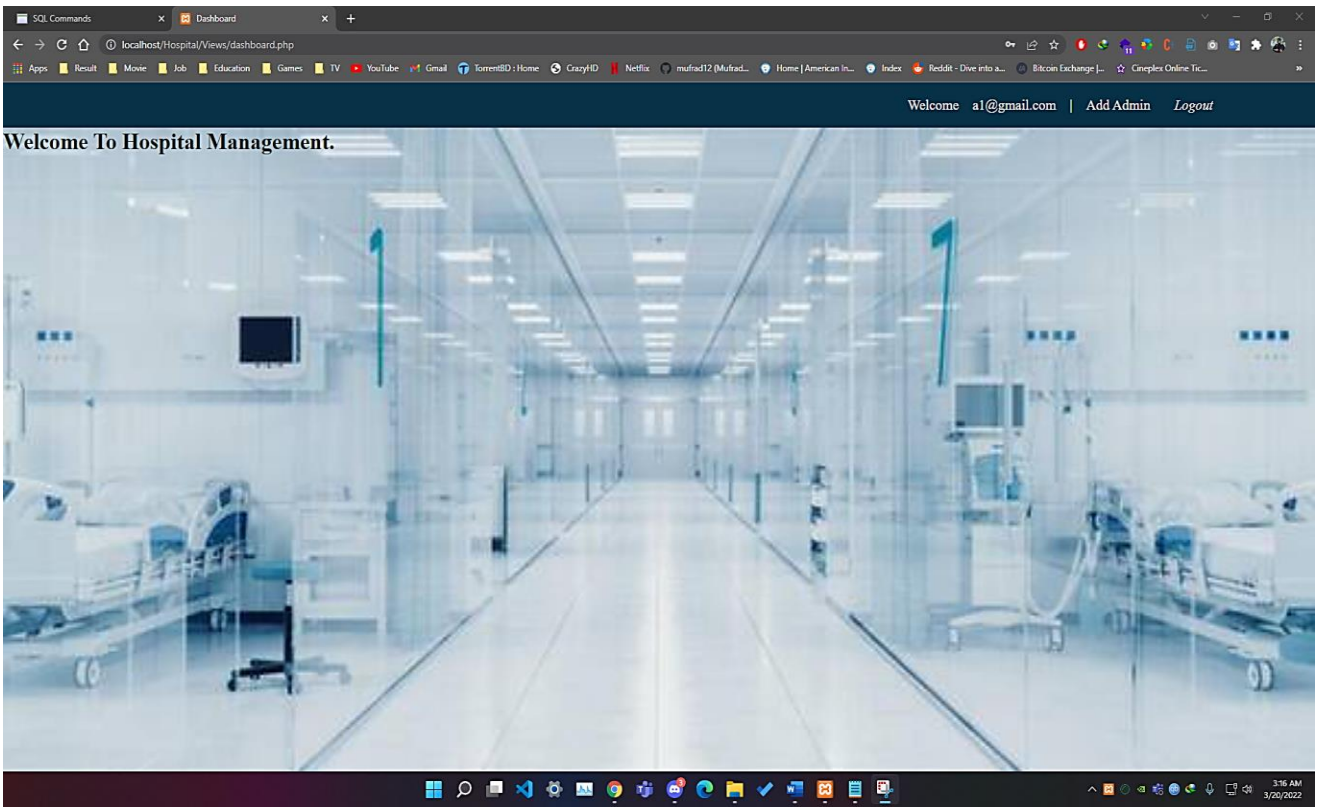
10. Create a program which will take the room_id of an room and will return the availability of the room

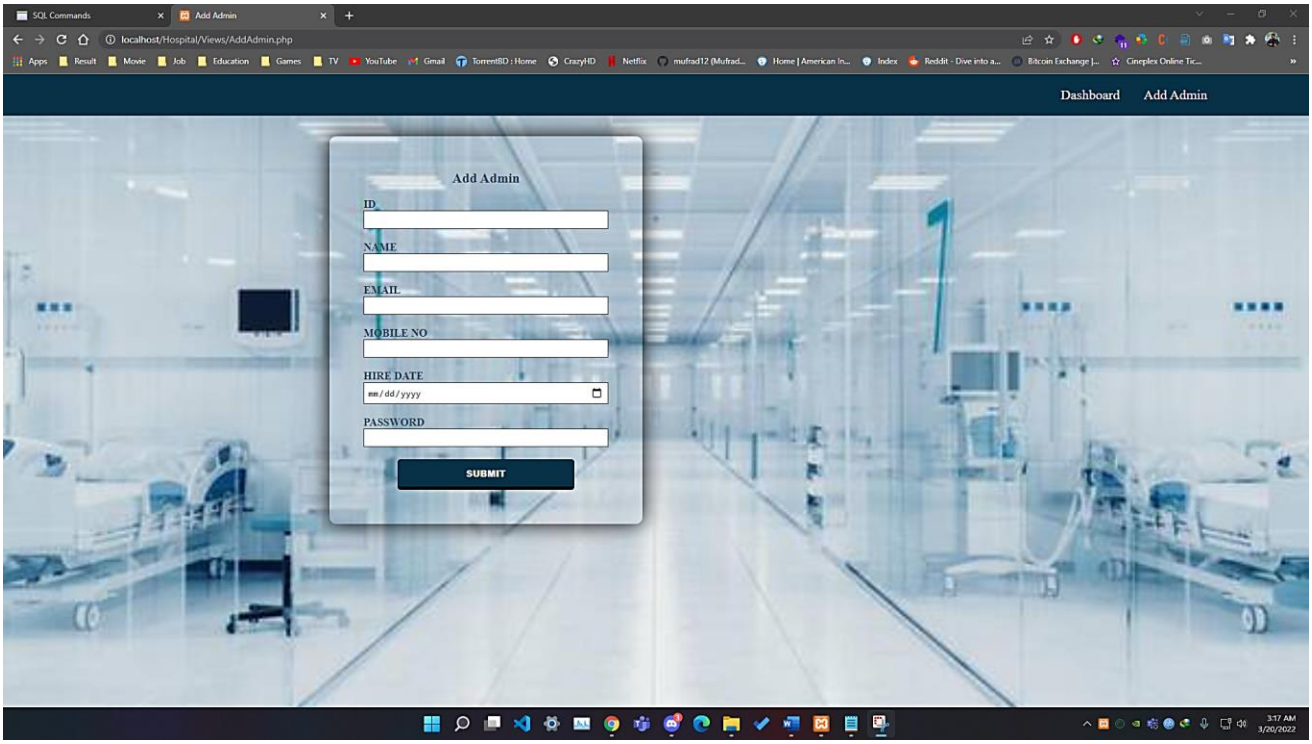
```
=> declare
rid room.room_id%type:=:Enter_room_id;
avl room.status%type;

begin
select status into avl from room where room_id = rid;

if(avl = 'available') then
DBMS_OUTPUT.PUT_LINE('Room no ' || rid || ' status: Available');

elsif(avl = 'not available') then
DBMS_OUTPUT.PUT_LINE('Room no ' || rid || ' status: Not available');

else
DBMS_OUTPUT.PUT_LINE('Room not found!');

end if;

end;
```



Interface


SQL Commands
Login
localhost/Hospital/Views/login.php
Home
Login
Hospital
Management
EMAIL
PASSWORD
Remember Me
LOGIN
Copyright@2022


SQL Commands
Dashboard
localhost/Hospital/Views/dashboard.php
Welcome a1@gmail.com | Add Admin Logout
Welcome To Hospital Management.


SQL Commands
Add Admin
localhost/Hospital/Views/AddAdmin.php
Dashboard Add Admin
Add Admin
ID
NAME
EMAIL
MOBILE NO
HIRE DATE
mm/dd/yyyy
PASSWORD
SUBMIT