Hospital Management System

Course: ADVANCED DATABASE MANAGEMENT SYSTEM
Section: B
Group: 8

Group Members:

Name	ID
AZWAD, AFID MD	18-38261-2
RIFAT UZ ZAMAN , MD.	18-38024-2
TONNI, RUBAIA AKTER	18-37969-2
MD. BADHON MIAH	18-38225-2
ABU BAKAR SIDDIQUE	18-38381-2

Submitted to:

JUENA AHMED NOSHIN

Assistant Professor
Faculty of Science & Technology
American International University-Bangladesh

Spring 2020-21

Contents

Introduction:

Hospitals are key institutions and there is a need for efficient service delivery in the hospital as good health is paramount to a happy society. As a result of this, there is a need for a system that will enable hospital management in making effective and efficient decisions. Recently, efforts are continuously being made in designing and constructing a user-friendly and reliable database system to satisfy hospital or medical management systems. On the other hand, many hospitals and medical centers are still adopting the manual system of hospital management. These methods of the medical management system have continued to pose a lot of setbacks and problems to medical practitioners, nurses, patients, and other staff in both government and private hospitals. Hospital management is an integrated hospital information system, which addresses all the major functional areas of multi-specialty hospitals. The hospital management enables better patient care, patient safety, patient confidentiality, efficiency, reduced costs, and a better management information system. It provides easy access to critical information thus enabling. The management to make better decisions on time. The advancement in innovative medical technology and global warming pose health hazards in most nations around the globe, especially developing countries. The traditional platform cannot any longer handle the mass medical data generated by the increasing world population. A platform that heavily relied on a traditional system where medical records kept in files and cabinets have become obsolete.

Project Proposal:

This project will automate the daily operations of ADMS hospital. The project keeps track of the staff and patient (in-patient, out-patient) details. It also takes care of the cabin, medical, invoice, and the doctor's appointment details. The system generates the daily cabin availability, the status of doctors. HOSPITAL MANAGEMENT is an integrated Hospital Information System, which addresses all the major functional areas of multi-specialty hospitals. The HOSPITAL MANAGEMENT enables better patient care, patient safety, patient confidentiality, efficiency, reduced costs, and better management information system. It provides easy access to critical information thus enabling the 45 management to take better decisions on time. This project deals with processing of each department in the hospital. This project sincerely aims to reduce the manual processing of each department. These details give the doctor, staff, specialists, and patient details including their salary, attendance, doctor’s appointments, and the billing system . The details of the Doctor and staff help the hospital to maintain the record of every person. Their attendance details help them to know about their attentive presence while salary is calculated. The billing system provides an efficient way for calculating bill details of the patients. In our project we have combined all features together. Here patients can see the specified doctor list, they can see the available word in the hospital. Patients can collect hospital bills and all other documents from the reception.

PROJECT OBJECTIVE

- To computerize all details regarding patient details & hospital details.
- To automate the process of cabin entries.
- To maintain records effectively.
- To manage status of staff and doctor availability.
- The project has information regarding the patient detail, Billing details and report.

FEATURES:

Patient Details:

It keeps track of all details about the patient. Patient id, patient name, address, admitted date, doctor name, room no in a form and stored for future reference. Also, patient details can be viewed in the table using a separate form with an attribute patient id.

Doctor Details:

It keeps track of all details about doctors of the hospital. Doctors name, id, address, qualification, cell no, e-mail is entered and stored in a separate form. Individual doctor details can be viewed in the table using a separate form with an attribute doctor id.

Staff Details:

It keeps track of all details about staff of the hospital. Staffs, Nurses name, staff id, address, qualification, cell no, e-mail are entered and stored in a separate form. Individual staff details can be viewed in the table using a separate form with an attribute Staff id.

Billing Details:

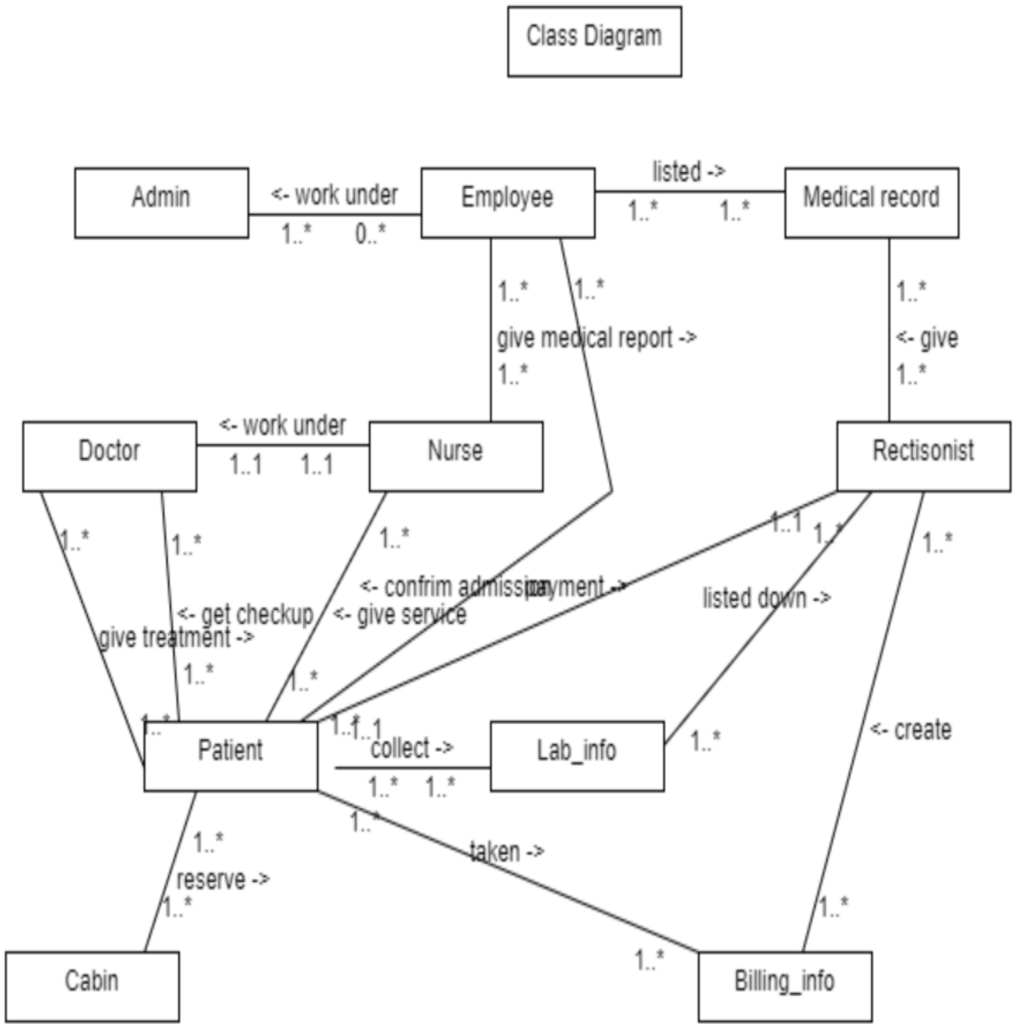
This module bills the both inpatient and outpatient who comes to hospital.

Cabin Details:

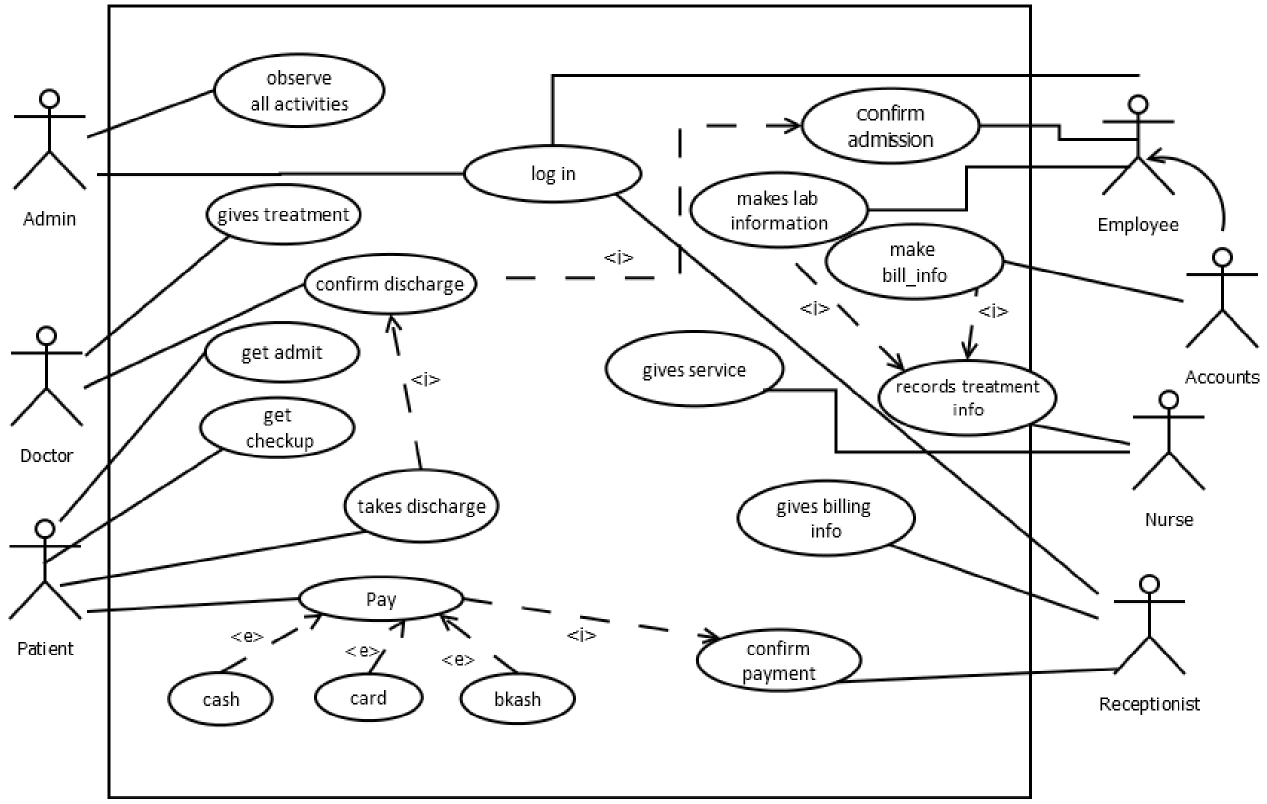
This module enters and stores the details about each cabin of the hospital for future reference. Individual cabin detail can be viewed in the table using cabin no. The attributes used in storing cabin name, floor no, no of rooms.

UML Diagrams:

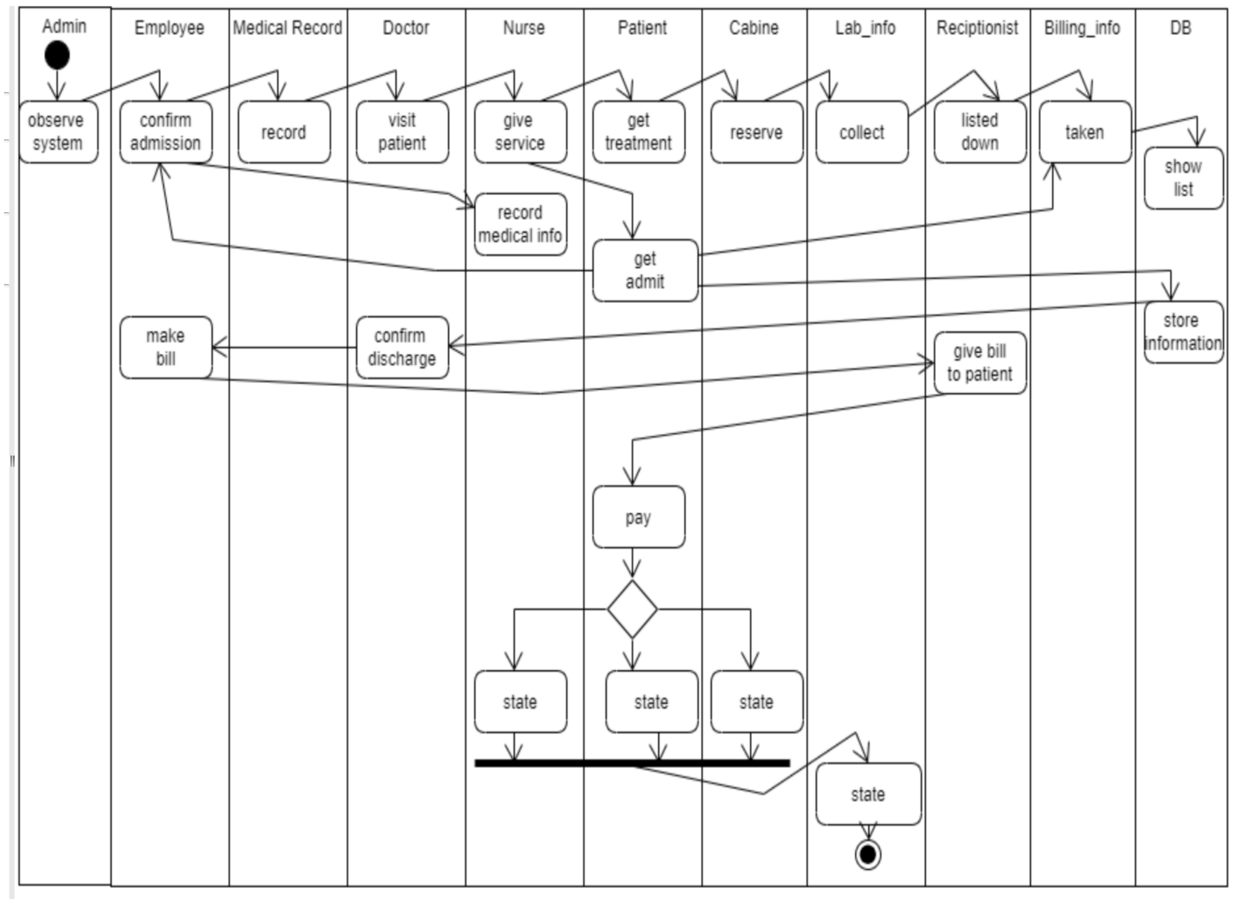
Class Diagram:



Use Case Diagram:

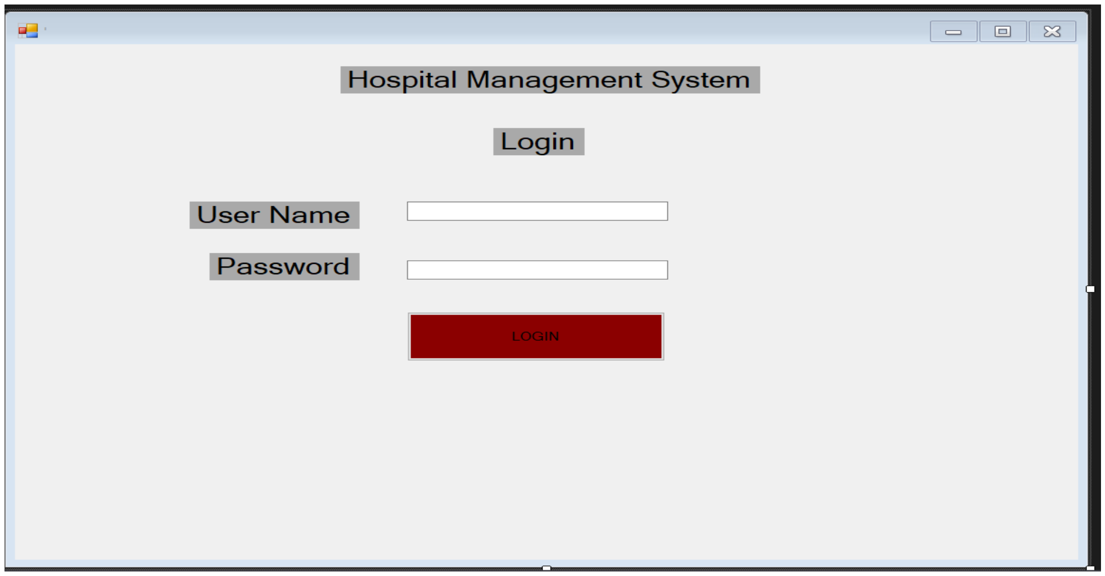


Activity Diagram:

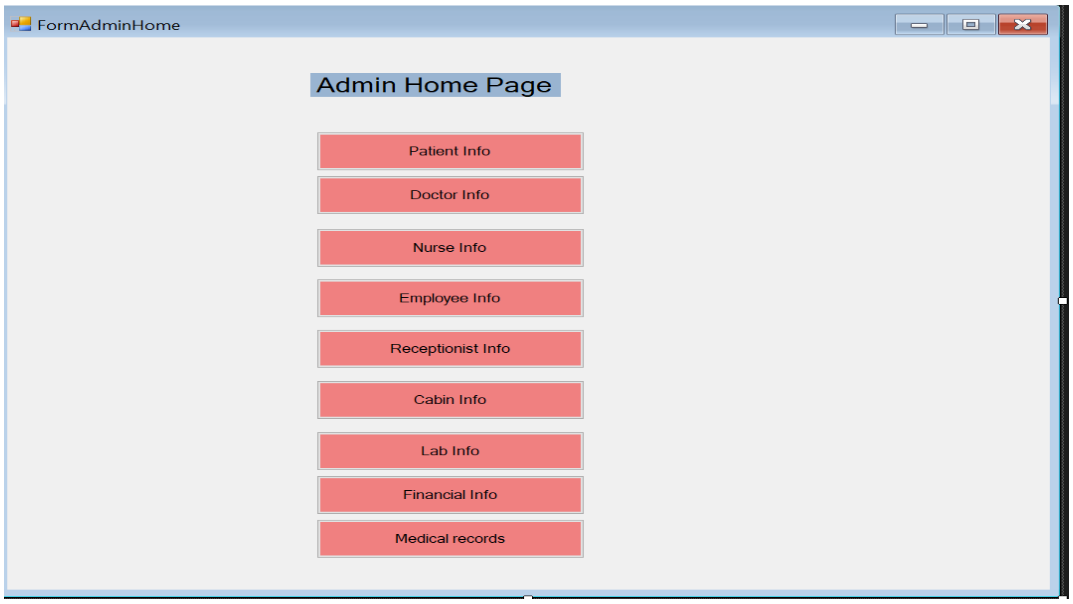


User Interface:

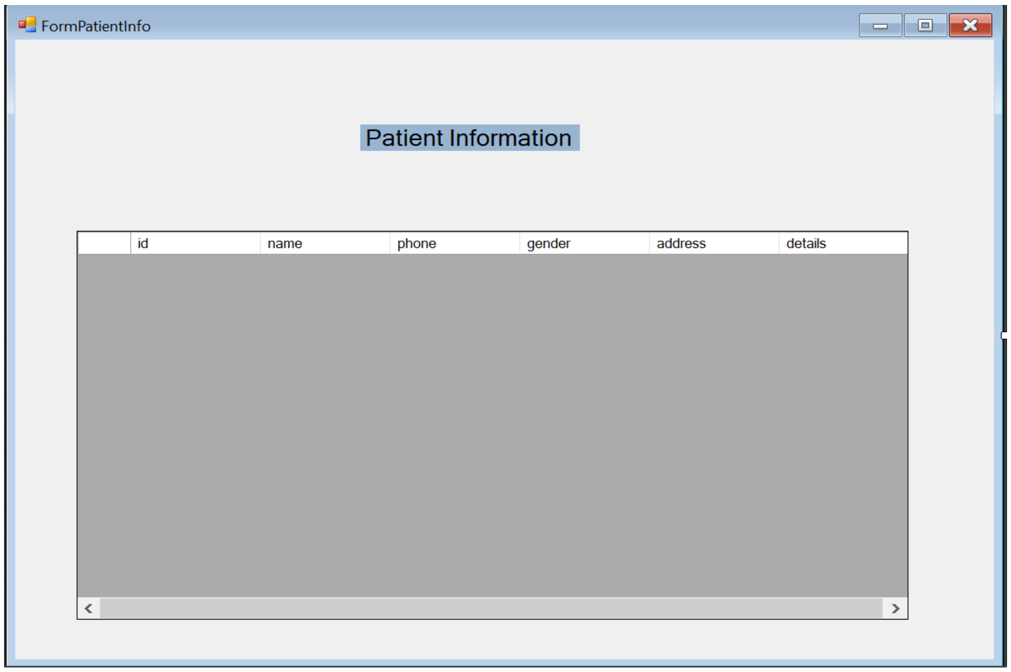
LOGIN UI:



Admin Home page UI:

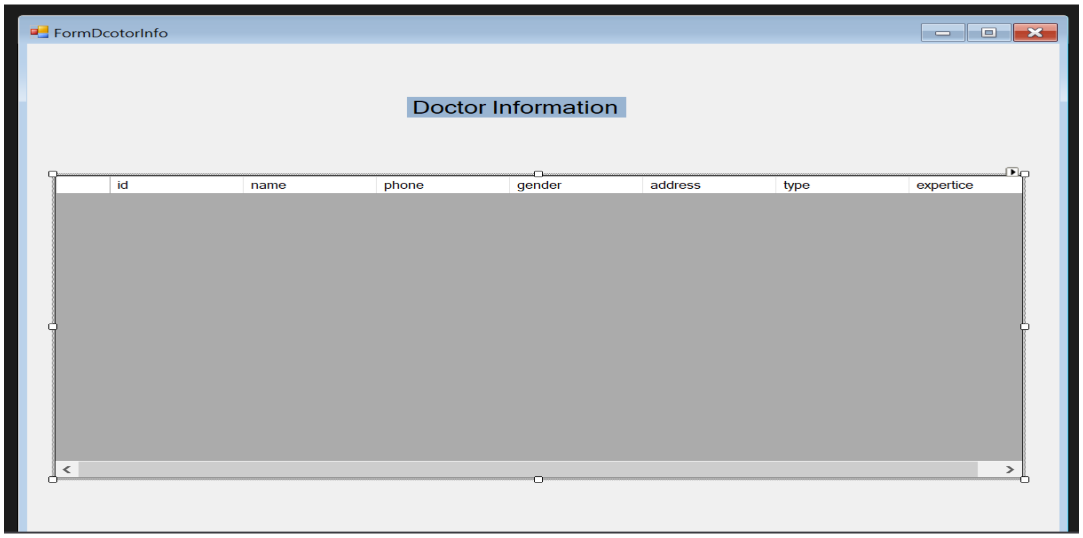


Patient Info UI:



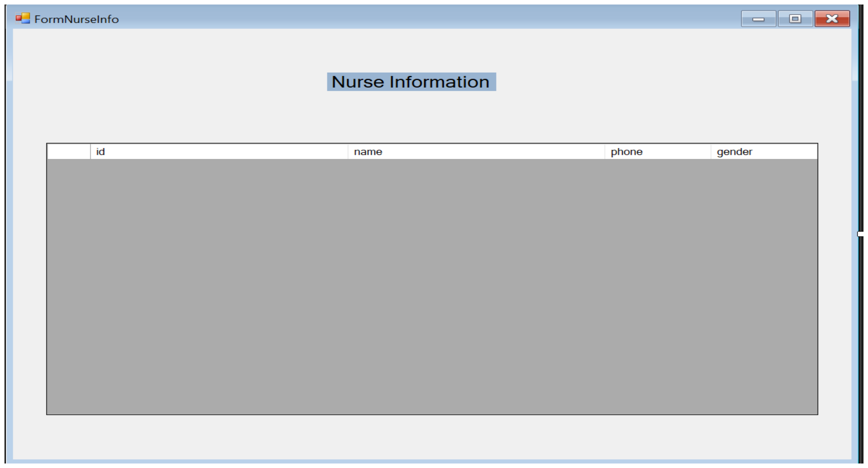
	id	name	phone	gender	address	details

Doctor information UI:



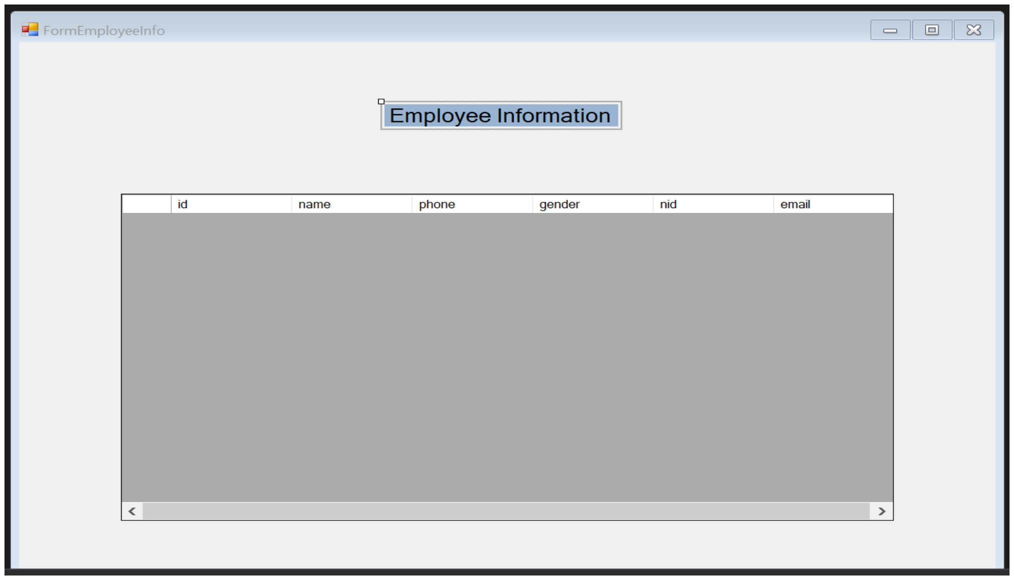
	id	name	phone	gender	address	type	expertice

Nurse Information UI:



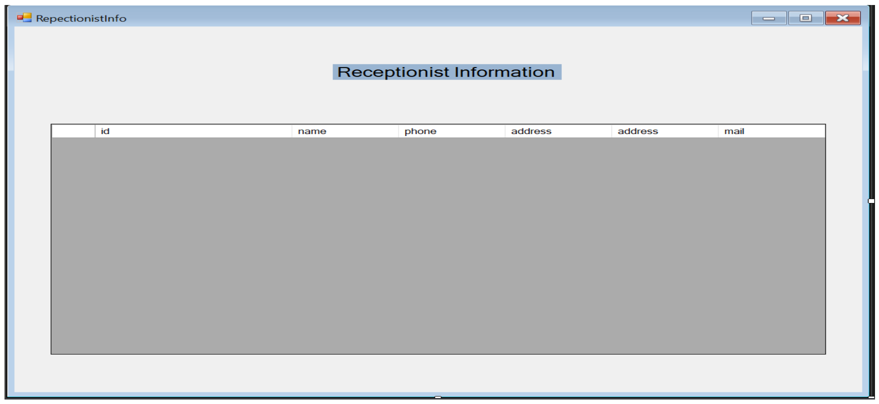
id	name	phone	gender

Employee Information UI:

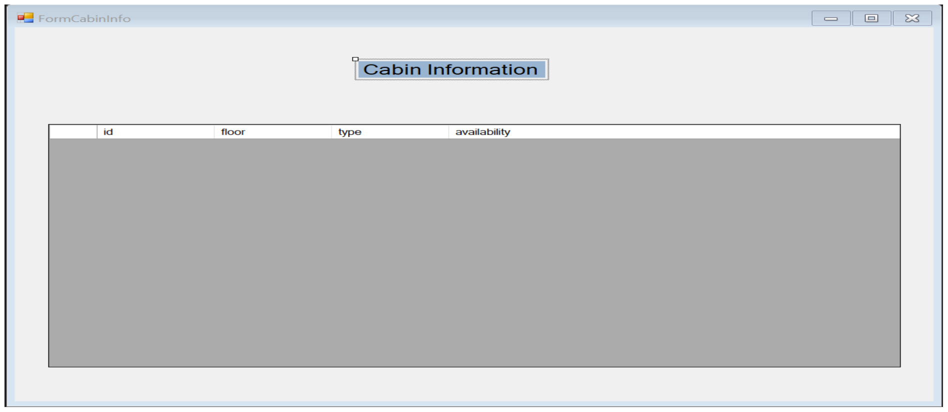


id	name	phone	gender	nid	email

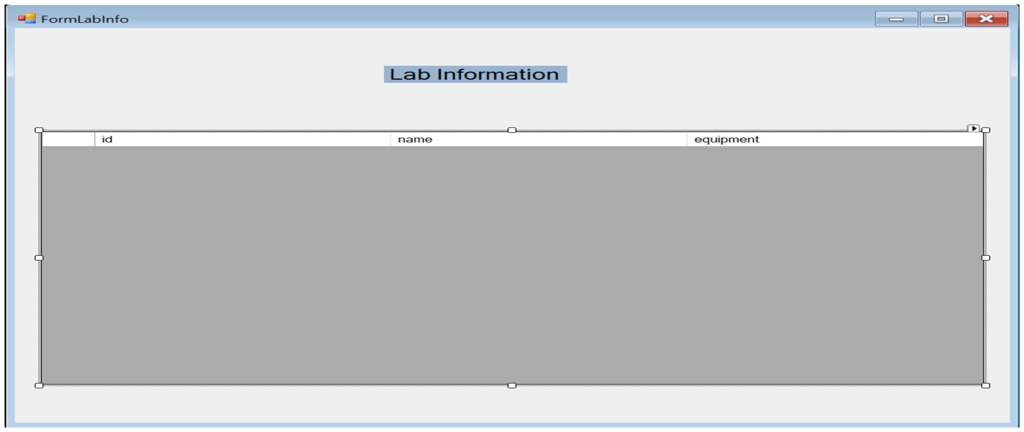
Receptionist Information UI:



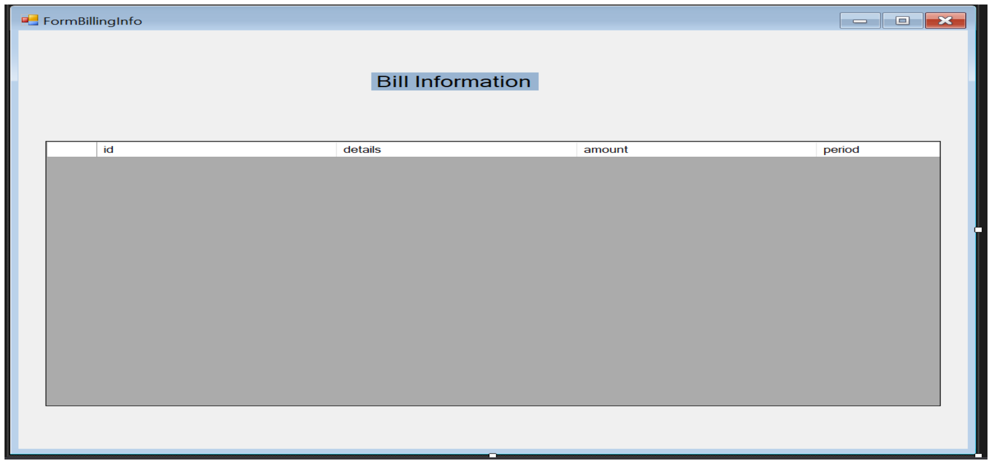
Cabin Information UI:



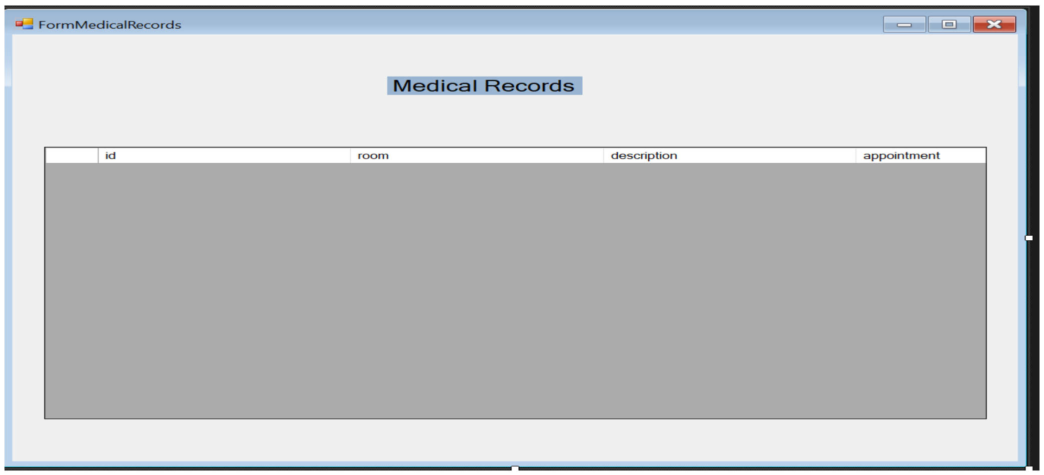
Lab Information UI:



Billing Information UI:



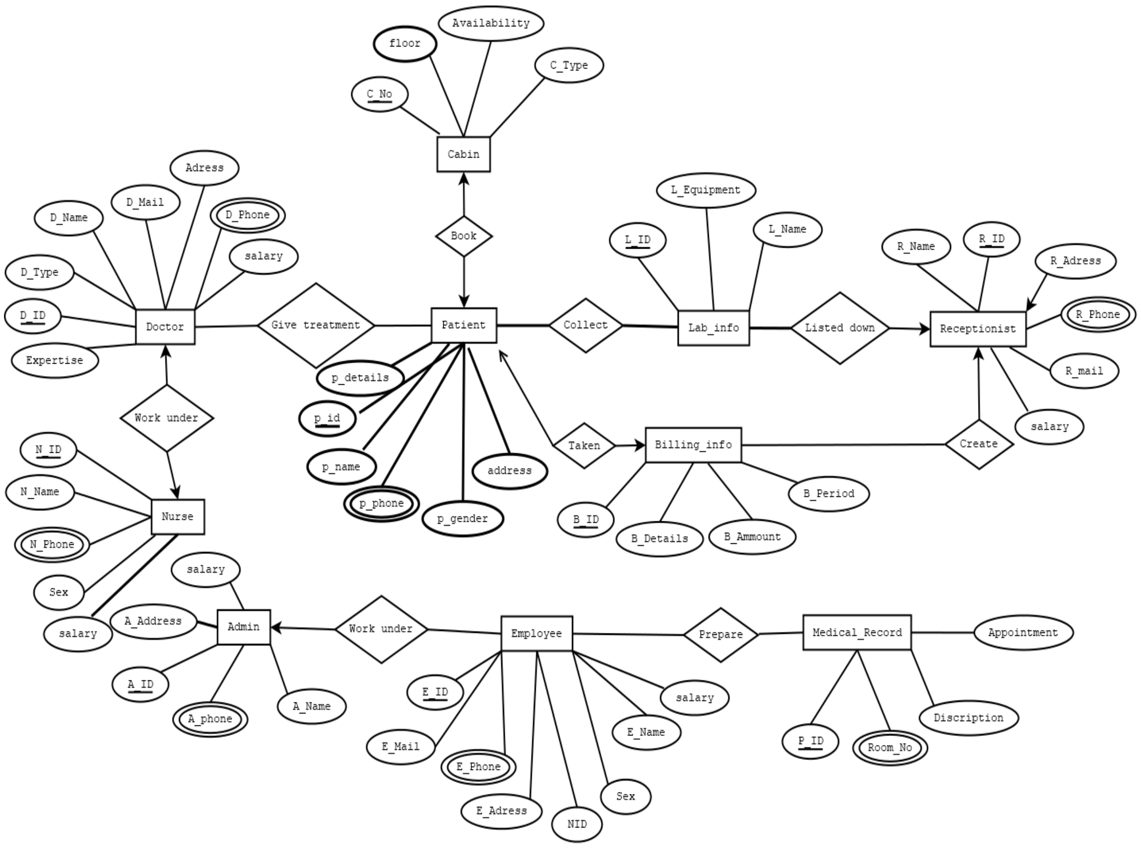
Medical Records UI:



Scenario Description:

ADMS hospital is a multi-specialty hospital that includes several doctors, patients, nurses, employees, receptionists working in the hospital. Each patient has a unique patient id. Patient data such as patient name, phone number, sex, address, details with admit and discharge is also stored in the system. Here Patients having different kinds of ailments come to the hospital and get checkup done from the concerned doctors. Also doctors give treatment as the patients need. Doctor has a specific id as well as name, address, mail, salary,phone number, expertise. There are different kinds of doctors in the hospital. Their types are also stored in a database. If a patient 's condition is not good, then they are admitted in the hospital and discharged after treatment. For this patient needs to book cabins. Cabin is also included by cabin no, floor, type, availability. There are nurses who work under the doctors. One doctor can have one or more nurses as an assistant. Each nurse has their own id. Nurse details like name, sex, salary, address, phone number are also stored in database. Hospital has an admin who observes all the activities of doctor, nurse, employees. Admin have specific id, name,salary, and password to enter the database. Doctors, Nurses, Employees and Receptionists work under the admin. One Admin has several employees. Each employee has their specific id. They also have address, name, sex,salary, NID, phone number. There is a receptionist in the reception with their different id number. Receptionist gives the lab information to the patients. Receptionists also have attributes like name, mail, salary, address, phone of their own. Receptionists release the billing info after a patient is discharged. Once patients are treated by the doctors, the employees of the accounts section in the hospital are responsible for calculating the medical fee, mailing bills, and receiving the payments. Billing info has its own id, treatment details, period, bill amount. Patient collects the billing info from the receptionist. Employees make the lab information for the patient. Lab information also specified by individual id, name, equipment. Employees also prepare medical records for the hospital purpose. Every medical record includes patient id, description, room number, appointment. This is how a hospital management system runs.

ER Diagram:



Normalization:

1 st

UNF:

d_id,d_type,d_email,d_name,address,d_phone,expertise,d_salary,p_id,p_name,p_phone,p_gender, address, p_details

1NF:

Phone number is a multivalued attribute.

1.d_id,d_type,d_email,d_name,address,d_phone,expertise,d_salary,p_id,p_name, p_phone,p_gender, address, p_details

2NF:

1. d_id,d_type,d_email,d_name,address,d_phone,expertise,d_salary

2. p_id,p_name,p_phone,p_gender,address, p_details

3NF:

1. d_id,d_type,d_email,d_name,address,d_phone,expertise,d_salary

2. p_id,p_name,p_phone,p_gender,address, p_details

Table Creation:

1. d_id,d_type,d_email,d_name,address,d_phone,expertise,d_salary

2. p_id,p_name,p_phone,p_gender,address, p_details

3. P_id,d_id

2 nd

UNF:

Work_under (d_id, d_name,d_type,d_email, address,d_phone, expertise,d_salary,n_id,n_name,n_phone,sex,n_salary)

1NF:

d_phone and n_phone are multivalued attributes.

1. d_id, d_name,d_type,d_email, address,d_phone, expertise,d_salary n_id,n_name,n_phone,sex,n_salary

2NF:

1. d_id, d_name,d_type,d_email, address,d_phone, expertise,d_salary

2. n_id,n_name,n_phone,sex,n_salary

3NF:

1. d_id, d_name,d_type,d_email, address,d_phone, expertise,d_salary

2. n_id,n_name,n_phone,sex,n_salary

Table Creation:

1. <u>d_id</u>, d_name,d_type,d_email, address,d_phone, expertise,d_salary **n_id**

2. <u>n_id</u>,n_name,n_phone,sex,n_salary

3 rd

UNF:

Book (p_id, p_name, d_phone ,p_sex, address,p_details,c_No,
floor,availability,c_type)

1NF:

p_phone is a multivalued attribute.

1. p_id, p_name, p_phone ,p_sex, address,p_details,c_no, floor,availability,c_type

2NF:

1. p_id, p_name, d_phone ,p_sex, address,p_details

2. <u>c_no</u>, floor,availability,c_type

3NF:

1. p_id, p_name, d_phone ,p_sex, address,p_details

2. c_No, Floor,Availability,c_type

Table Creation:

1. <u>p_id</u>, p_name, d_phone ,p_sex, address,p_details, **c_no**

2. <u>c_no</u>, floor,availability,c_type

UNF:

Collect (p_id, p_name, p_phone ,p_gender, address,p_details,l_id,l_name, l_equipment)

1NF:

p_phone is a multivalued attribute.

1. p_id, p_name, p_phone ,p_gender, address,p_details,l_id,l_name, l_equipment

2NF:

1. p_id, p_name, p_phone ,p_gender, address,p_details

2. l_id,l_name, l_equipment

3NF:

1. p_id, p_name, p_phone ,p_gender, address,p_details

2. l_id,l_name, l_equipment

Table Creation:

1. <u>p_id,</u> p_name, p_phone ,p_gender, address,p_details

2. <u>l_id</u>,l_name, l_equipment

3. **<u>p_id, l_id</u>**

UNF:

Taken (p_id, p_name, p_phone ,p_gender, address,p_details,b_id, b_details, b_amount,b_period)

1NF:

p_phone is a multivalued attribute.

1. p_id, p_name, p_phone ,p_gender, address,p_details, b_id, b_details, b_amount,b_period

2NF:

1. p_id, p_name, p_phone ,p_gender, address,p_details

2. b_id, b_details, b_amount,b_period

3NF:

1. p_id, p_name, p_phone ,p_gender, address,p_details

2. b_id, b_details, b_amount,b_period

Table Creation:

1. <u>p_id</u>, p_name, p_phone ,p_gender, address,p_details

2. <u>b_id</u>, b_details, b_amount,b_period, **p_id**

6 th

UNF:

Listed_down (r_id, r_name, address ,r_phone ,r_email,r_salary ,l_id, l_name,l_equipment)

1NF:

r_phone is a multivalued attribute.

1. r_id, r_name, address ,r_phone ,r_email ,r_salary,l_id, l_name,l_equipment

2NF:

1. r_id, r_name, address ,r_phone ,r_email,r_salary

2. l_id, l_name,l_equipment

3NF:

1. r_id, r_name, address ,r_phone ,r_email,r_salary

2. l_id, l_name,l_equipment

Table Creation:

1. r_id, r_name, address ,r_phone ,r_email,r_salary

2. l_id, l_name,l_equipment ,**r_id**

7 th

UNF:

Create(r_id, r_name, address ,r_phone ,r_email ,r_salary,b_id, b_details,b_amount,b_period)

1NF:

r_phone is a multivalued attribute.

1. r_id, r_name, address ,r_phone ,r_email ,r_salaryb_id, b_details,b_amount,b_period

2NF:

1. r_id, r_name, address ,r_phone ,r_email,r_salary

2. b_id, b_details,b_amount,b_period

3NF:

1. r_id, r_name, address ,r_phone ,r_email,r_salary

2. b_id, b_details,b_amount,b_period

Table Creation:

1. r_id, r_name, address ,r_phone ,r_email,r_salary

2. b_id, b_details,b_amount,b_period, **r_id**

8 th

UNF:

Under (a_id, a_name,a_phone,a_address ,a_salary,e_id,e_name, e_email,address,e_phone,e_NId,e_sex,e_salary)

1NF:

a_phone and e_phone are multivalued attributes.

1. a_id, a_name,a_phone,a_address ,a_salary,e_id,e_name, e_email,address,e_phone,e_NId,e_sex,e_salary

2NF:

1. a_id, a_name,a_phone,a_address,a_salary

2. e_id,e_name, e_email,address,e_phone,e_NId,e_sex,e_salary

3NF:

1. a_id, a_name,a_phone,a_address,a_salary

2. e_id,e_name, e_email,address,e_phone,e_NId,e_sex,e_salary

Table Creation:

1. <u>a_id,</u> a_name,a_phone,a_address,a_salary

2.<u>e_id</u>,e_name, e_email,address,e_phone,e_NId,e_sex,e_salary,**a_id**

9 th

UNF:

Prepare(e_id,e_name,,e_email,e_phone,e_address,e_id,e_sex,e_phone,e_salary,p_id,Room_no,description,

appointment)

1NF:

e_phone is a multivalued attribute.

1.e_id,e_name,,e_email,e_phone,e_address,e_id,e_sex,e_phone,e_salary,p_id,Room_no,description, appointment

2NF:

1 e_id,e_name,,e_email,e_phone,e_address,e_id,e_sex,e_phone,e_salary

2. p_id,Room_no,description,appointment

3NF:

1. e_id,e_name,,e_email,e_phone,e_address,e_nid,e_sex,e_phone,e_salary

2. p_id,Room_no,description,appointment

Table Creation:

1. e_id, e_name,,e_email,e_phone,e_address,e_nid,e_sex,e_salary

2. p_id,Room_no,description,appointment

3. **e_id,p_id**

TEMPORARY TABLE

1. ~~d_id,d_type,d_email,d_name,address,d_phone,expertise,d_salary~~

2. ~~p_id,p_name,p_phone,p_gender,address, p_details~~

3. **p_id**,**d_id**

4. d_id, d_name,d_type,d_email, address,d_phone, expertise,d_salary **n_id**

5. n_id,n_name,n_phone,sex,n_salary

6. p_id, p_name, p_phone ,p_sex, address,p_details, **c_no**

7. c_no, floor,availability,c_type

8. ~~p_id, p_name, p_phone ,p_gender, address,p_details~~

9. ~~l_id,l_name, l_equipment~~

10. **p_id, l_id**
11. ~~p_id, p_name, p_phone ,p_gender, address,p_details~~
12. b_id, b_details, b_amount,b_period, **p_id**
13. r_id, r_name, address ,r_phone ,r_email,r_salary
14. l_id, l_name,l_equipment ,**r_id**
15. ~~r_id, r_name, address ,r_phone ,r_email,r_salary~~
16. b_id, b_details,b_amount,b_period, **r_id**
17. a_id, a_name,a_phone,a_address,a_salary
18. e_id,e_name, e_email,address,e_phone,e_nid,e_sex,e_salary,**a_id**
19. ~~e_id, e_name,,e_email,e_phone,e_address,e_nid,e_sex,e_salary~~
20. p_id,room_no,description,appointment
21. **e_id,p_id**

Final table:

1. **p_id**,**d_id**
2. d_id, d_name,d_type,d_email, address,d_phone, expertise,d_salary, **n_id**
3. n_id,n_name,n_phone,sex,n_salary
4. p_id, p_name, p_phone ,p_sex, address,p_details, **c_no**
5. c_no, floor,availability,c_type
6. **p_id**, **l_id**
7. b_id, b_details, b_amount,b_period, **p_id**
8. r_id, r_name, address ,r_phone ,r_email,r_salary

9. l_id, l_name,l_equipment ,**r_id**

10. b_id, b_details,b_amount,b_period, **r_id**

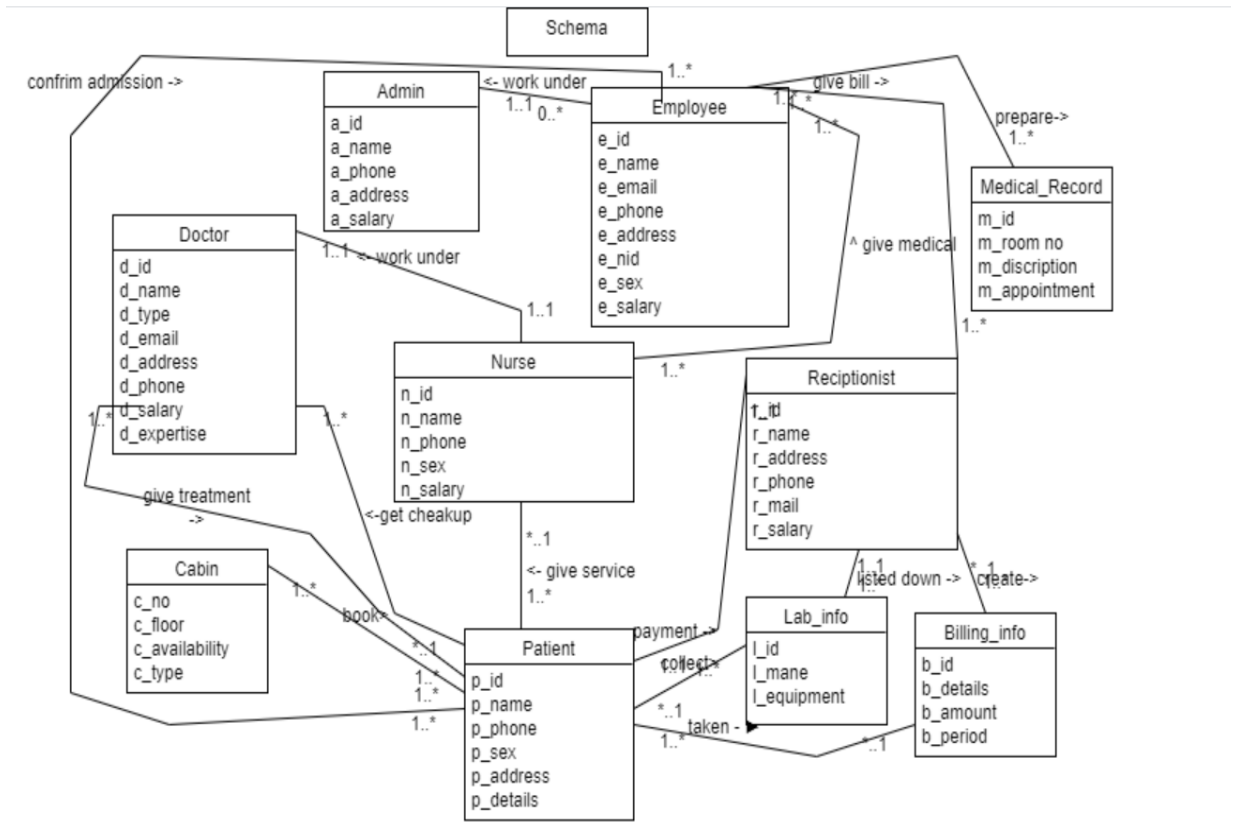
11. a_id, a_name,a_phone,a_address,a_salary

12. e_id,e_name, e_email,address,e_phone,e_nid,e_sex,e_salary,**a_id**

13. p_id,room_no,description,appointment

14. **e_id, p_id**

Schema Diagram:



User Creation

create user Hospital identified by patient;

grant connect, resource, unlimited tablespace to Hospital;

ALTER USER HOSPITAL DEFAULT TABLESPACE USERS;

ALTER USER HOSPITAL TEMPORARY TABLESPACE TEMP;

Table Creation:

1. CREATE TABLE Nurse (n_id NUMBER(20) CONSTRAINT PK_Nurse PRIMARY KEY,n_name VARCHAR2(50), n_phone NUMBER(20), sex VARCHAR2(50),n_salary NUMBER(20));

2. CREATE TABLE Cabin (c_no NUMBER(20) CONSTRAINT PK_Cabin PRIMARY KEY, floor NUMBER(20), availability VARCHAR2(50), c_type VARCHAR2(50));

3. CREATE TABLE Reception (r_id NUMBER(20) CONSTRAINT PK_Reception PRIMARY KEY,r_name VARCHAR2(50), address VARCHAR2(50), r_phone NUMBER(20), r_email VARCHAR2(50),r_salary NUMBER(20));

4. CREATE TABLE Admin (a_id NUMBER(20) CONSTRAINT PK_Admin PRIMARY KEY,a_name VARCHAR2(50), a_phone NUMBER(20), a_address VARCHAR2(50),a_salary NUMBER(20));

5. CREATE TABLE Report (p_id NUMBER(20) CONSTRAINT PK_Report PRIMARY KEY,room_no NUMBER(30), description VARCHAR2(50), appoinment VARCHAR2(50));

6. CREATE TABLE Doctor (d_id NUMBER(20) CONSTRAINT PK_Doctor PRIMARY KEY, d_name VARCHAR2(50), d_type VARCHAR2(50), d_email VARCHAR2(50), address VARCHAR2(50),d_phone NUMBER(30), expertise VARCHAR2(50), d_salary NUMBER(20),n_id NUMBER(20) CONSTRAINT FK_n_id REFERENCES Nurse);

7. CREATE TABLE Patient (p_id NUMBER(20) CONSTRAINT PK_Patient PRIMARY KEY, p_name VARCHAR2(50), p_phone NUMBER(30), p_sex VARCHAR2(50), address VARCHAR2(50), p_details VARCHAR2(50),c_no

NUMBER(20) CONSTRAINT FK_c_no REFERENCES Cabin);

8. CREATE TABLE Patient_Bill (b_id NUMBER(20) CONSTRAINT PK_Patient_Bill PRIMARY KEY, b_details VARCHAR2(50), b_amount NUMBER(30), b_period VARCHAR2(50), p_id NUMBER(20) CONSTRAINT FK_p_id REFERENCES Patient);

9. CREATE TABLE Reception_Bill (b_id NUMBER(20) CONSTRAINT PK_Reception_Bill PRIMARY KEY, b_details VARCHAR2(50), b_amount NUMBER(30), b_period VARCHAR2(50), r_id NUMBER(20) CONSTRAINT FK_r_id REFERENCES Reception);

10. CREATE TABLE Lab (l_id NUMBER(20) CONSTRAINT PK_Lab PRIMARY KEY, l_name VARCHAR2(50), l_equipmemt VARCHAR2(50), r_id NUMBER(20) CONSTRAINT FK__r_id REFERENCES Reception);

11. CREATE TABLE Employee (e_id NUMBER(20) CONSTRAINT PK_Employee PRIMARY KEY, e_name VARCHAR2(50), e_email VARCHAR2(50), e_phone NUMBER(30), e_address VARCHAR2(50),_e_nid NUMBER(20), e_sex VARCHAR2(50),_e_salary NUMBER(20), a_id NUMBER(20) CONSTRAINT FK_a_id REFERENCES Admin);

12. CREATE TABLE Patient_doctor (p_id number(20),d_id number(20),FOREIGN KEY (d_id) REFERENCES doctor(d_id),FOREIGN KEY (p_id) REFERENCES patient(p_id), primary key(p_id,d_id));

13.CREATE TABLE Patient_lab (p_id number(20),l_id number(20),FOREIGN KEY (l_id) REFERENCES lab(l_id),FOREIGN KEY (p_id) REFERENCES patient(p_id), primary key(p_id,l_id));

14.CREATE TABLE Patient_Employee (p_id number(20),e_id number(20),FOREIGN KEY (e_id) REFERENCES employee(e_id),FOREIGN KEY (p_id) REFERENCES patient(p_id), primary key(p_id,e_id));

<u>Value Insertion</u>

INSERT INTO Nurse VALUES(10,'Sid',01799098765, 'male',4000);

INSERT INTO Nurse VALUES(11,'jiu',01889098765, 'female',5000);

INSERT INTO Nurse VALUES(12,'july',01789098765, 'female',4000);

```
INSERT INTO Nurse VALUES(13,'Riu',01989098765, 'female',4000);
INSERT INTO Nurse VALUES(14,'Poly',01689098765, 'female',5000);



INSERT INTO Cabin VALUES(101,22,'jiu', '1st');
INSERT INTO Cabin VALUES(106,23,'july', '2nd');
INSERT INTO Cabin VALUES(103,24,'Riu', '1st');
INSERT INTO Cabin VALUES(104,25,'Sid', '2nd');
INSERT INTO Cabin VALUES(105,26,'jiu', '1st');

INSERT INTO Reception VALUES(51,'Jim','Dhaka', 01899098769, 'jim@gmail.com',10000);
INSERT INTO Reception VALUES(52,'Ron','Dhaka', 01999098769, 'ron@gmail.com',12000);
INSERT INTO Reception VALUES(55,'Rim','Dhaka', 01799098769, 'rim@gmail.com',10000);
INSERT INTO Reception VALUES(56,'Sun','Dhaka', 01699098769, 'sun@gmail.com',12000);
INSERT INTO Reception VALUES(57,'Rif','Dhaka', 01888098769, 'rif@gmail.com',10000);

INSERT INTO Admin VALUES(71,'Ruz',01656743212, 'Dhaka',60000);
INSERT INTO Admin VALUES(72,'Euz',01956743212, 'Sylhet',50000);
INSERT INTO Admin VALUES(73,'RMun',01756743212, 'Dhaka',60000);
INSERT INTO Admin VALUES(74,'Herd',01906743212, 'Sylhet',50000);
INSERT INTO Admin VALUES(75,'Jerry',01616743212, 'Dhaka',60000);

INSERT INTO Report VALUES(31,212,'DgjEK', 'No');
INSERT INTO Report VALUES(32,221,'DgdEK', 'Yes');
INSERT INTO Report VALUES(33,213,'DEK', 'No');
INSERT INTO Report VALUES(34,223,'EK', 'Yes');

INSERT INTO Report VALUES(35,224,'DK', 'No');


INSERT INTO Doctor VALUES(1,'Cedric','MBBS', 'ced@gmail.com', 'Dhaka',01799098765,
'jjjj',10,10);

INSERT INTO Doctor VALUES(2,'Codrak','Phd', 'cod@gmail.com', 'Rajshahi',01899098765,
'kkkk',11,11);

INSERT INTO Doctor VALUES(3,'Medric','MBBS', 'med@gmail.com', 'Dhaka',01399098765,
'jjjj',10,12);

INSERT INTO Doctor VALUES(4,'Rodrak','Phd', 'rod@gmail.com', 'Rajshahi',01299098765,
'kkkk',11,13);

INSERT INTO Doctor VALUES(5,'Sodrak','Phd', 'sod@gmail.com', 'Natore',01499098765,
'kkkk',11,14);


INSERT INTO Patient VALUES(21,'Joy', 01799098765,'male', 'Dhaka', 'aaaa',101);

INSERT INTO Patient VALUES(22,'Joty', 01899098765,'female', 'Bogra', 'bbbb',102);

INSERT INTO Patient VALUES(23,'Roy', 01999098765,'male', 'Dhaka', 'ssaa',103);

INSERT INTO Patient VALUES(24,'Soty', 01699098765,'female', 'Bogra', 'yybb',104);

INSERT INTO Patient VALUES(25,'Toy', 01709098765,'male', 'Dhaka', 'yyaa',105);


INSERT INTO Patient_Bill VALUES(31,'ECG', 765 , 'Moring',21);

INSERT INTO Patient_Bill VALUES(32,'MRI', 705 , 'Night',22);

INSERT INTO Patient_Bill VALUES(33,'ECG', 765 , 'Moring',23);

INSERT INTO Patient_Bill VALUES(34,'MRI', 706 , 'Night',24);

INSERT INTO Patient_Bill VALUES(35,'ECG', 7656 , 'Moring',25);


INSERT INTO Reception_Bill VALUES(31,'ECG', 765 , 'Moring',51);
INSERT INTO Reception_Bill VALUES(32,'MRI', 705 , 'Night',52);

INSERT INTO Reception_Bill VALUES(36,'ECG', 767 , 'Moring',55);

INSERT INTO Reception_Bill VALUES(34,'MRI', 707 , 'Night',56);

INSERT INTO Reception_Bill VALUES(35,'ECG', 766 , 'Moring',57);


INSERT INTO Lab VALUES(61,'DEK', 'Thatoskop',51);

INSERT INTO Lab VALUES(62,'REK', 'Thatoskop1',52);

INSERT INTO Lab VALUES(63,'MEK', 'Thatoskop2',57);

INSERT INTO Lab VALUES(64,'WEK', 'Thatoskop3',55);

INSERT INTO Lab VALUES(65,'KEK', 'Thatoskop4',56);


INSERT INTO Employee VALUES(200,Azwad, 'zaman@gmail.com', 01799098765,
'Dhaka',09875467, 'male',12000,71);

INSERT INTO Employee VALUES(201,'Badhon', 'badhon@gmail.com', 01999098765,
'Khulna',76575467, 'male',1000, 72);

INSERT INTO Employee VALUES(202,'Tonni', 'tonni@gmail.com', 01899098765,
'Dhaka',098754443, 'male',15000,73);

INSERT INTO Employee VALUES(203,'Rifat', 'ruz@gmail.com', 01979098765,
'Khulna',7653467, 'male',16000, 74);

INSERT INTO Employee VALUES(204,'Salim', 'karim@gmail.com', 01399098765,
'Dhaka',09555467, 'male',12000,75);


INSERT INTO Patient_doctor VALUES(21,1);

INSERT INTO Patient_doctor VALUES(22,2);

INSERT INTO Patient_doctor VALUES(23,3);

INSERT INTO Patient_doctor VALUES(24,4);
INSERT INTO Patient_doctor VALUES(25,5);


INSERT INTO Patient_lab VALUES(21,61);

INSERT INTO Patient_lab VALUES(22,62);

INSERT INTO Patient_lab VALUES(23,63);

INSERT INTO Patient_lab VALUES(24,64);

INSERT INTO Patient_lab VALUES(25,65);



INSERT INTO Patient_Employee VALUES(21,200);

INSERT INTO Patient_Employee VALUES(22,201);

INSERT INTO Patient_Employee VALUES(23,202);

INSERT INTO Patient_Employee VALUES(24,203);

INSERT INTO Patient_Employee VALUES(25,204);
```

Query Writing:

SQL

single-row functions:

i) Show all the information of the patient whose name is joy

Select * from Patient where p_name=’Joy’;

ii) Show all the information of the Employee whose name is Azwad

Select * from Employee where e_name = 'Azwad';

iii) Show all the information of the Admin whose name is Herd

Select * from Admin where a_name = 'Herd';

group function:

i) Show the maximum bill of all types of bill details.

Select max(b_amout) from Patient_Bill group by (b_details);

ii) Show the maximum salary of the employees group by their address.

Select max(e_salary) from Employee group by e_address;

iii) Show the maximum salary of the Nurses group by their sex.

Select max(n_salary) from Nurse group by sex;

Subquery:

i) Show the patient info who pays the bill 765

Select * from patient where p_id=(select p_id from Patient_Bill where b_amount = 705);

ii) Show the Receptionist info who is dealing with the bill amount 707

Select * from Reception where r_id=(select r_id from Reception_Bill where b_amount = 707);

iii) Show the Admin info who is dealing with the employee ‘Azwad’

Select * from Admin where a_id=(select a_id from Employee where e_name = 'Azwad');

Joining:

i) Show the patient id name and their bill amount.

Select p.p_id,p.p_name,b.b_amount from Patient p, Patient_Bill b where p.p_id=b.p_id;

ii) Show the bill id, bill period, patient id and their bill amount.

Select r.b_id, r.b_period, b.p_id, b.b_amount from Reception_Bill r, Patient_Bill b where r.b_id=b.b_id;

iii) Show the patient id name with their cabin informations

Select p.p_id,p.p_name , c.c_no, c.floor, c_type from Patient p, cabin c where c.c_no= p.c_no;

views:

i) Create a view with the patient id name and their bill amount.

Create view patient_Bill_Info as Select p.p_id,p.p_name,b.b_amount from Patient p, Patient_Bill b where p.p_id=b.p_id;

ii) Create a view to show receptionist id, name and salary.

```
Create view Receptionist as Select r_id,r_name,r_salary from Reception;
```

iii) Create a view to show Employee's id, name, phone number and salary.

```
Create view Employee_info as Select e_id, e_name,e_phone, e_salary from Employee;
```

<u>Synonym</u>:

i) Create a synonym for patient_Bill_Info as patient_info

```
Create SYNONYM patient_info for patient_Bill_Info
```

ii) Create a synonym for Reception_Bill to short the name

```
Create SYNONYM Bills for Reception_Bill;
```

iii) Create a synonym for Cabin as rooms

```
Create SYNONYM rooms for cabin;
```

PL/SQL

<u>function</u> -

i) create a pl/sql function that has a parameter that takes an id of a doctor and returns his/her salary.

```
DECLARE
  salary number;

FUNCTION display(id in number)
RETURN number
IS
   sal number;
```

```
BEGIN
   select d_salary into sal from doctor where d_id=id;
   return sal;
END;
BEGIN

   salary := display(2);
   dbms_output.put_line(' Salary '|| salary);
END;
```

ii) create a pl/sql function that returns the maximum salary of the nurse.

```
DECLARE
   salary number;

FUNCTION maxSal
RETURN number
IS
    sal number;
BEGIN
   select max(n_salary) into sal from nurse ;
   return sal;
END;
BEGIN

   salary := maxSal;
   dbms_output.put_line('Max Salary '|| salary);
END;
```

iii) create a pl/sql function that returns the average bill from the patient bill table.

```
DECLARE
   bill number;

FUNCTION avargeBill
RETURN number
IS
    a number;
BEGIN
   select avg(b_amount) into a from patient_bill ;
   return a;
END;
BEGIN

   bill := avargeBill;
   dbms_output.put_line('Avg bill '|| bill);
END;
```

procedure:

i) create a pl/sql procedure that has a parameter that take patient id and display the cabin no of the patient .

```
DECLARE
 c number;

PROCEDURE information(id in number) IS
BEGIN
  select c_no into c from patient where p_id=id;
     dbms_output.put_line('Cabin no '|| c);

END;
BEGIN
   information(21);

END;
```

ii) create a pl/sql procedure that has 2 parameters first one is admin id and second one is new salary and update the salary.

```
DECLARE
PROCEDURE updateSalary(id in number, sal in number)IS
BEGIN
  update admin set a_salary=sal where a_id=id ;
END;
BEGIN
   updateSalary(71,1000);
```

```
END;
```

iii) create a pl/sql procedure that has a parameter that take employee id and display the name of employee .

```
DECLARE
 name varchar2(50);


PROCEDURE empName(id in number) IS
BEGIN
  select e_name into name from employee where e_id=id;
     dbms_output.put_line('Employee name '|| name);

END;
BEGIN
   empName(201);

  END;
```

Record:

i) Create a record that can output the name of the doctor whose id is 2.

```
declare
doctor_rec doctor%rowtype;
begin
select * into doctor_rec from doctor
where d_id=2;
dbms_output.put_line(doctor_rec.d_id||' '||doctor_rec.d_name||' '||doctor_rec.d_type ||'
'||doctor_rec.d_email ||' '||doctor_rec.d_phone ||' '||doctor_rec.d_salary);
End
```

ii)Create a record that can output the name of all the doctors.

```
declare
doctor_rec doctor%rowtype;
begin
for doctor_rec
in(select * from doctor)
loop
dbms_output.put_line(doctor_rec.d_name);
end loop;
End
```

iii) Create a record that can output all the bill information.

```
declare
bill_rec patient_bill%rowtype;
begin
for bill_rec
in(select * from patient_bill)
loop
dbms_output.put_line(bill_rec.b_id || ' '|| bill_rec.b_details   || ' '|| bill_rec.b_amount  || ' '||
bill_rec.b_period  || ' '|| bill_rec.p_id);
end loop;
end
```

<u>Cursor:</u>

i) Create a cursor that can output the id and name of all the patients.

```
declare
name patient.p_name%type;
```

```
id patient.p_id%type;
cursor c_patient is
select p_id,p_name from patient;
begin
open c_patient;
loop
fetch c_patient into id,name;
exit when c_patient%notfound;
dbms_output.put_line(id||' ' ||name);
end loop;
close c_patient;
End
```

ii) Create a cursor that can output the name of all the labs.

```
declare
name lab.l_name%type;

cursor c_lab is
select l_name from lab;
begin
open c_lab;
loop
fetch c_lab into name;
exit when c_lab%notfound;
dbms_output.put_line(' Lab Name ' ||name);
end loop;
close c_lab;
End
```

iii)Create a cursor that can output the id and their salary for all of the nurses.

```
declare
sal nurse.n_salary%type;
id nurse.n_id%type;

cursor c_nurse is
select n_id, n_salary from nurse;
begin
```

```
open c_nurse;
loop
fetch c_nurse into id,sal;
exit when c_nurse%notfound;
dbms_output.put_line(id || ' get salary '||sal);
end loop;
close c_nurse;
End
```

Package:

ii) Create a package that contains a doctor which can display the name and salary is passed as its parameter

```
----- Create or specifying a package -------

    CREATE OR REPLACE PACKAGE doctor_pack AS

      PROCEDURE display_d_name(d_id doctor. d_no%type);

      PROCEDURE display_d_salary(d_id doctor. d_no%type);

    END doctor _pack;


    ---------  Package Body ---------

    CREATE OR REPLACE PACKAGE BODY doctor _pack AS

     PROCEDURE display_d_name(d_id doctor. d_no%TYPE) IS

      d_nam doctor.d_name%TYPE;

      BEGIN

        SELECT d_name INTO d_nam

        FROM doctor

        WHERE d_no = d_id;

        dbms_output.put_line('Doctor  Name: '|| d_nam);

      END display_d_name;
```

```
PROCEDURE display_d_salary(d_id doctor. d_no%TYPE) IS
  d_sal doctor.salary%TYPE;
  BEGIN
    SELECT salary INTO d_sal
    FROM doctor
    WHERE d_no = d_id;
    dbms_output.put_line('Doctor  Salary '|| d_sal);
  END display_d_salary;
END doctor_pack;
/
-------- Using the package -----
BEGIN
doctor_pack.display_d_name('11');
doctor_pack.display_d_salary('12');
END;
```

ii) Create a package that contains an employee which can display the name and salary as its parameter.

```
----- Create or specifying a package -------
    CREATE OR REPLACE PACKAGE employee _pack AS
      PROCEDURE display_e_name(e_id employee. e_no%type);
      PROCEDURE display_e_salary(e_id employee. e_no%type);
    END employee _pack;

    ---------  Package Body ---------

    CREATE OR REPLACE PACKAGE BODY employee _pack AS
```

```
  PROCEDURE display_e_name(e_id employee. e_no%TYPE) IS
   e_nam employee.e_name%TYPE;
   BEGIN
      SELECT e_name INTO e_nam
      FROM employee
      WHERE e_no = e_id;
      dbms_output.put_line('Employee Name: '|| e_nam);
   END display_e_name;

PROCEDURE display_e_salary(e_id employee. e_no%TYPE) IS
   e_sal employee.salary%TYPE;
   BEGIN
      SELECT e_salary INTO e_sal
      FROM employee
      WHERE e_no = e_id;
      dbms_output.put_line('Employee Salary '|| e_sal);
   END display_e_salary;
END employee _pack;
/
-------- Using the package -----
BEGIN
employee _pack.display_e_name('200');
employee _pack.display_e_salary('202');
```

```
      END;
```

iii) Create a package that contains a nurse which can display the name is passed as its parameter

```
      ----- Create

or specifying a package -------

      CREATE OR

      REPLACE PACKAGE nurse_pack AS

        PROCEDURE

      display_n_name(n_id nurse. n_no%type);

      END nurse

      _pack;


      ---------   Package Body

      ---------

      CREATE OR

      REPLACE PACKAGE BODY nurse _pack AS

        PROCEDURE

      display_n_name(n_id nurse. n_no%TYPE) IS

        n_nam

      nurse.n_name%TYPE;

        BEGIN

          SELECT n_name INTO n_nam

          FROM doctor

          WHERE n_no = n_id;
          dbms_output.put_line('Nurse  Name: '|| n_nam);
        END
      display_n_name;
       END
      nurse_pack;
      /
      --------
      Using the package -----
      BEGIN
      nurse_pack.display_n_name('110');
      END
```

Trigger:

i) Create a trigger in such a way that whenever a new row is inserted into the lab table an output 'INSERTED IN LAB' is generated.

```
CREATE OR REPLACE TRIGGER new_insert_in_lab
after INSERT ON Lab
FOR EACH ROW
WHEN (NEW.l_id > 0)
BEGIN
  dbms_output.put_line(' INSERTED IN LAB ');
END;
/
select * from Lab;
```

```
INSERT INTO Lab VALUES(66,'KEKE', 'Thatoskop5',57);
```

ii) Create a row-level trigger for the Doctor table that would get executed by the DML statement like UPDATE, INSERT or DELETE on that table.

```
CREATE OR REPLACE TRIGGER display_d_salary_changes

BEFORE DELETE OR INSERT OR UPDATE ON doctor

FOR EACH ROW

WHEN (NEW.d_id > 0)

DECLARE

  sal_diff number;

BEGIN

  sal_diff := :NEW.d_salary  - :OLD.d_salary;

  dbms_output.put_line('Old salary: ' || :OLD.d_salary);

  dbms_output.put_line('New salary: ' || :NEW.d_salary);

  dbms_output.put_line('Salary difference: ' || sal_diff);

END;

/

        update doctor set d_salary='5000' where d_id='10'

INSERT INTO Doctor VALUES(11,'Cedric','MBBS', 'ced@gmail.com', 'Dhaka',01799098765,
'jjjj',10,10);

            select * from doctor;

 drop trigger display_d_salary_changes
```

iii) Create a trigger in such a way that whenever a new row is inserted into the employee table an output 'New Row Added' is generated.

```
CREATE OR REPLACE TRIGGER new_row_added

after INSERT ON Employee

FOR EACH ROW

WHEN (NEW.e_id > 0)

BEGIN


  dbms_output.put_line('NEW ROW ADDED ');


END;

/

select * from Employee;

        INSERT INTO Employee VALUES(203,'Tomy', 'tomy@gmail.com', 01399098765,
'Dhaka',0987541113, 'male',15900,74);
```

Conclusion:

The project Hospital Management System (HMS) is for computerizing the working in a hospital. The software takes care of all the requirements of an average hospital and is capable of providing easy and effective storage of information related to patients that come up to the hospital. It generates test reports; provide prescription details including various tests, check-up and lab info prescribed to patients and doctors. It also provides lab details and billing facilities. The system also provides the facility of backup as per the requirement.