“HOSPITAL MANAGEMENT SYSTEM”

Advanced DBMS Project report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

Ankur Pal (Univ. Roll. No. 200013139137)

Ashish Kumar (Univ. Roll. No. 200013139138)

Submitted to

Dr. Jasvant Singh

FACULTY OF ENGINEERING AND TECHNOLOGY

UNIVERSITY OF LUCKNOW, LUCKNOW

2022-2023

DECLARATION

We hereby declare that this submission is our work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material that to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Name: Ankur Pal

Roll No.:200013139137

Date:

Signature:

Name: Ashish Kumar

Roll No.:200013139138

Date:

Signature:

CERTIFICATE

This is to certify that the Project Report entitled with "**Hospital Management System**" which is submitted by **Ankur Pal** (**200013139137**) and **Ashish Kumar** (**200013139138**) partial fulfilment of the requirement for the award of degree B. Tech. in Department of **Computer Science and Engineering** of **Faculty of Engineering And Technology, Lucknow University, U.P., Lucknow**, is a record of the candidates own work carried out by them under my supervision. The matter embodied in this Project report is original and has not been submitted for the award of any other degree.

Dr. Jasvant Singh
Advance DBMS Project In charge.

Er. Zeeshan Siddique
Incharge of theDepartment.

ABSTRACT

Our project Hospital Management system includes registration of patients, storing their details into the system, and also booking their appointments with doctors.

Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. User can search availability of a doctor and the details of a patient using the id. The Hospital Management

System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

It is having mainly two modules. One is at Administration Level and other one is of user I.e. of patients and doctors. The Application maintains authentication in order to access the application. Administrator task includes managing doctors information, patient's information. To achieve this aim a database was designed one for the patient and other for the doctors which the admin can access. The complaints which are given by user will be referred by authorities.

The Patient modules include checking appointments, prescription. User can also pay doctor's Fee online.

ACKNOWLEDGEMENTS

I am greatly indebted to my guide **Dr. Jasvant Singh** for her/his invaluable guidance during the course of the mini project. She always gave useful suggestions and also helped us when the work was not moving ahead at times.

Ankur Pal
Ashish Kumar
University Of Lucknow

TABLE OF CONTENTS

LIST OF FIGURES USED IN THE PROJECT

LIST OF TABLES USED IN THE PROJECT

Problem Statement

In this busy world we don't have the time to wait in infamously long hospital queues

HMS will help us overcome all these problems because they can check whether the doctor they want to meet is available or not. Doctors can also confirm or decline appointments, this help both patient and the doctor because if the doctor declines' appointment then patient will know this in advance and patient will visit hospital only when the doctor confirms' the appointment this will save time and money of the patient.

HMS is essential for all healthcare establishments, be it hospitals, nursing homes, health clinics, rehabilitation centers, dispensaries, or clinics. The main goal is to computerize all the details regarding the patient and the hospital. The installation of this healthcare software results in improvement in administrative functions and hence better patient care, which is the prime focus of any healthcare unit.

Benefits of implementing a hospital management system:

- ***Role-Based Access Control*** o Allows employees to access only the necessary information to effectively perform their job duties
    - Increases data security and integrity

- ***Overall cost reduction*** o Cuts down paper costs as all the data are computerized o No separate costs for setting up physical servers

- ***Data accuracy*** o Removes human errors
    - Alerts when there's a shortage of stock

- ***Data security*** o Helps to keep patients records private o Restricts access through role-based access control

- ***Revenue management***
    - Makes daily auditing simple
    - Helps with statistics and other financial aspects

CHAPTER 1

INTRODUCTION

1.1 PURPOSE

This software will help the company to be more efficient in registration of their patients and manage appointments, records of patients. It enables doctors and admin to view and modify appointments schedules if required. The purpose of this project is to computerize all details regarding patient details and hospital details.

1.2 SCOPE

The system will be used as the application that serves hospitals, clinic, dispensaries or other health institutions. The intention of the system is to increase the number of patients that can be treated and managed properly.
If the hospital management system is file based, management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also could be misplaced by losing data and information.

1.3 DEFINITIONS, ACRONYMS, and ABBREVIATIONS

1. **Cardiologist** - treats heart disease.
2. **Rheumatologist**- diagnoses and treats arthritis and other immune-related diseases and conditions
3. **Otolaryngology**- the ears, nose, and throat.
4. **Psychiatrist** - treats patients with mental and emotional disorders. **5.** Ophthalmologist - treats eye defects, injuries, and diseases

6. Neurologist- treats nerves and treats their diseases

- **SRS**: Software Requirement Specification.
- **DFD:** Data Flow Diagram.
- **ENT**- Ear, Nose and Throat Specialist.
- **BG** - Blood group
- **Appt –** Appointment.
- **Sign up -** Creating New User.
- **Log in -** Logging in Existing User.

- **PhNo** - Mobile number.
- **Addr** – Address.
- **Expr** – Experience.

1.4 OVERVIEW

Our application contains two modules – the admin module and the user module. Our application will not only help the admin to preview the monthly and/or yearly data but it will also allow them to edit, add or update records. The software will also help the admin to monitor the transactions made by the patients and generate confirmations for the same. The admin will be able to manage and update information about doctors.
The user module can be accessed by both the doctors and the patients. The doctor can confirm and/or cancel appointments. The doctors can even add prescriptions for their patients using our application. The patients will be able to apply for the appointment and make transaction for the same, and can even cancel appointments with the doctors. They can track details about the previous transactions made by them.

Advantages

- The system automates the manual procedure of managing hospital activities.
- Doctors can view their patients' treatment records and details easily.
- It even generates an instant bill.
- The system is convenient and flexible to be used.
- It saves their time, efforts, money and resources.

Disadvantages

- Requires large database.
- The admin has to manually keep updating the information by entering the details in the system.
- Need Internet connection.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Product Perspective

This Hospital Patient Info Management System is a self-contained system that manages activities of the hospital.
Due to improperly managed details medical center faces quite a lot of difficulties in accessing past data as well as managing present data. The fully functional automated hospital management system which will be developed through this project will eliminate the disadvantages caused by the manual system by improving the reliability, efficiency and performance. The usage of a database to store patient, employee, stock details etc. will accommodate easy access, retrieval, and search and manipulation of data. The access limitations provided through access privilege levels will enhance the security of the system. The system will facilitate concurrent access and convenient management of activities of the medical center.

2.1.1 System Interfaces

- ***User Interfaces***
    - This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.
- ***Hardware Interfaces***
    - **Laptop/Desktop PC**-Purpose of this is to give information when Patients ask information about doctors, medicine available lab tests etc. To perform such Action it need very efficient computer otherwise due to that reason patients have to wait for a long time to get what they ask for.
    - **Laser Printer (B/W)** - This device is for printing patients' info etc.
- ***Software Interfaces***
    - **Mysql server** - Database connectivity and management
    - **OS Windows 7/8/8.1**- Very user friendly and common OS
    - **Python**

2.1.2 System Specifications

2.1.2.1 H/W Requirement

- Core i5 processor 2GB Ram.
- 20GB of hard disk space in terminal machines
- 1TB hard disk space in Server Machine

2.1.2.2 S/W Requirement

- Windows 7 or above operating system
- Mysql server

2.2 Product functions

- Provide access to registered users only.
- Registration of new patients. o Enable patient to view their record. o Enable patient to update their record. o Generate appointment date and timing.
- Confirmation by doctor. o Patients can do Payment. o Modification in schedule by patient. o Admin access to patient's record.
- Admin Verify Payment and Generate Bill/Receipt.
- Admin can view monthly/yearly records.

2.3 DATA FLOW DIAGRAM (DFD)

CONTEXT LEVEL DIAGRAM

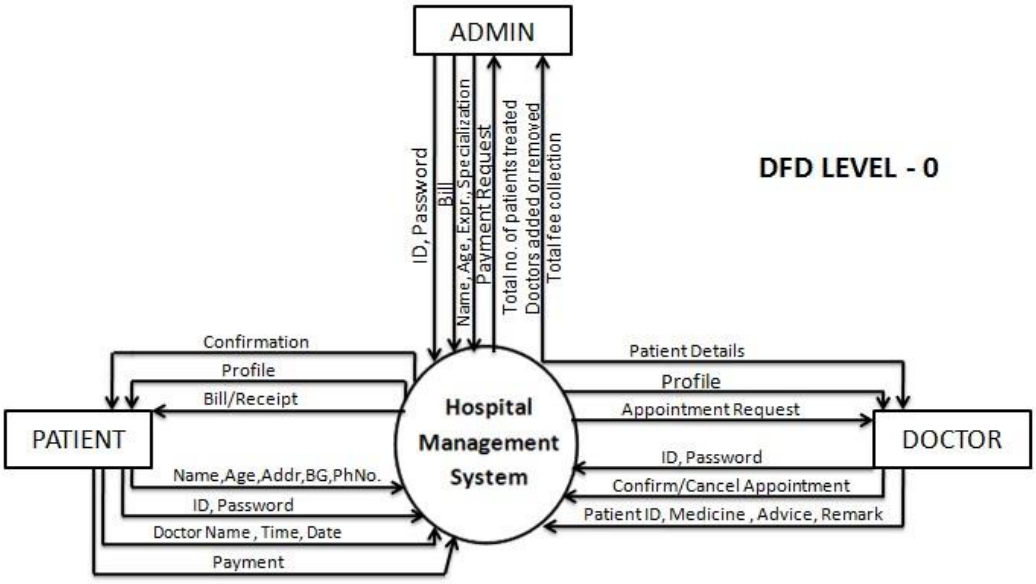

FIGURE 2.1 CONTEXT LEVEL DFD

DFD LEVEL – 1

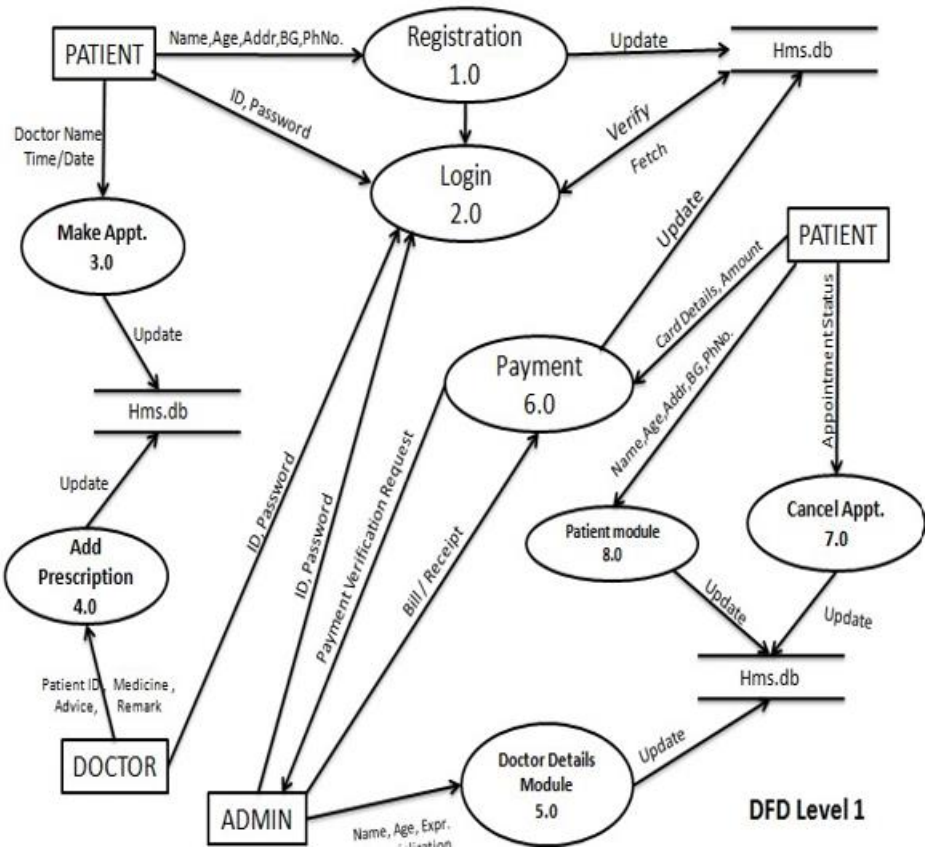

FIGURE 2.2 DFD Level 1

2.4 USE CASE DIAGRAM

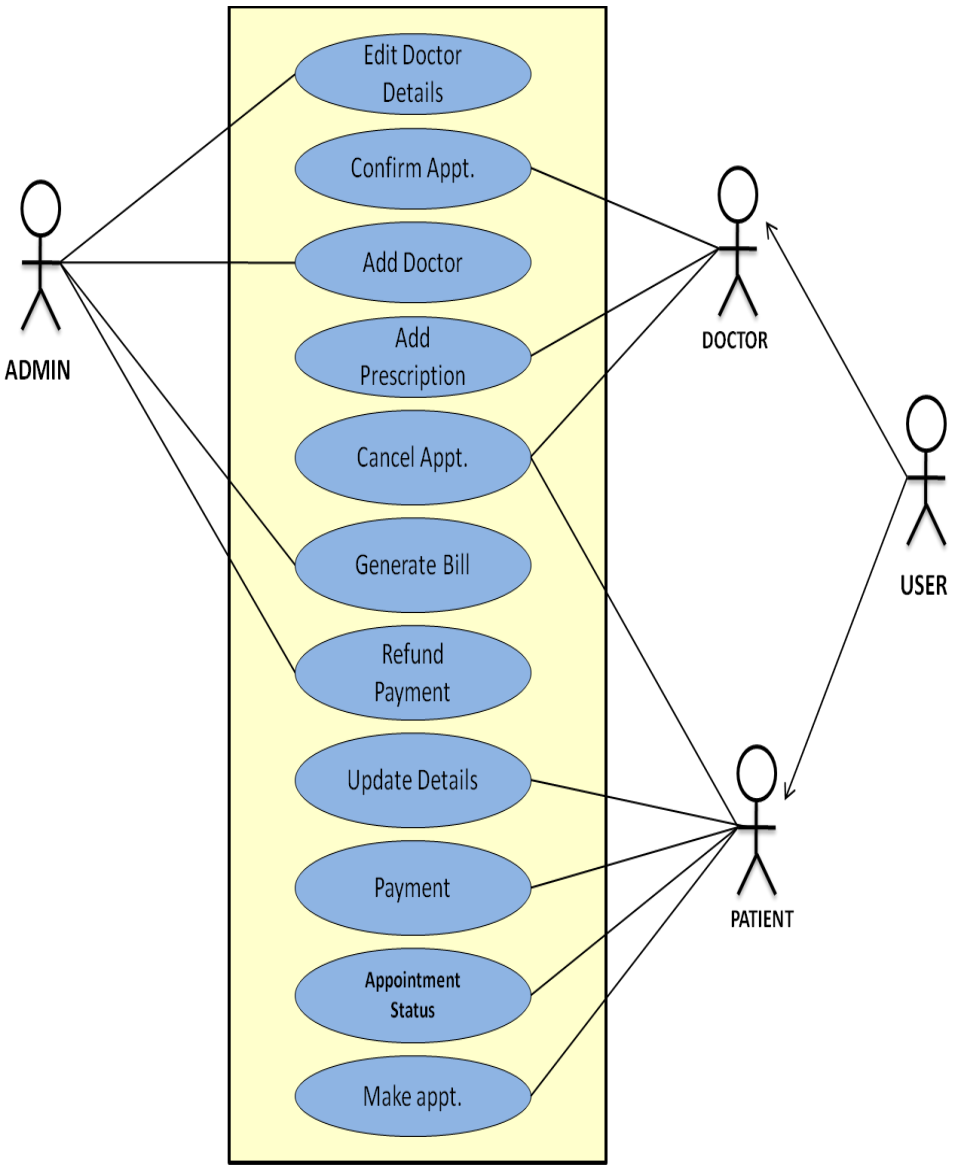

FIGURE 2.3 USE CASE DIAGRAM

2.5 USE CASE DESCRIPTION

(1) PATIENT

*** REGISTRATION**

DESCRIPTION - The new patient can register themselves and add their details like name, age , gender, blood group etc. The patient entry will be made in the hms database.

PRE -CONDITION – The patient must be a new patient, If necessary fields left by user then prompt user to fill the necessary fields.

MAIN FLOW OF EVENTS
1. Patient selects sign up in login module. 2. A registration form get displayed
3. Patient fills the required details.

POST CONDITIONS - Patient record is added to hms database.

*** UPDATION**

DESCRIPTION-The patient should be enabled to update his/her details and the changes should reflect in hms database.

PRE-CONDITION – The patient must be a registered patient, The patient cannot update details after treatment starts.

MAIN FLOW OF EVENTS

1. Patient logs in to the system.
2. Patient view his record
3. Patient selects update details.
4. Now patient may change the necessary fields.
5. Pop of update details.

POST CONDITION - The record of patient is updated in hms database.

(2) DOCTOR

DESCRIPTION- The doctor view patient record/ update his details and add description of the treatment given to patient.

PRE-CONDITION – The doctor must be a registered doctor, System does not allow the doctor to modify the qualification, hospital managed details.

MAIN FLOW OF EVENTS

1.Doctor logs in to the system.

2. Doctor may select view patient.

2.1 Patient record is displayed with treatment history.

3. Doctor add description of patient treatment.

4. Doctor may select appointment details

4.1 Appointment Requests is displayed with schedule.

5. Doctor confirm or cancel appointment.

(3) ADMIN

DESCRIPTION - The admin add doctor, update docotr details and verify payment and generate Bill/Reciept for the same.

MAIN FLOW OF EVENTS

1. Admin logs in the system.

2. Admin may add doctor new doctor.

2.1 admin fills the doctor's details.

3. Admin view Doctor record.

3.1 Admin enters the doctor id in the system.

3.2 Doctor details are displayed, Admin can update details.

4. Admin Verify the payment submited by the Patient.

4.1 Generate Bill/Reciept and confirmation message for the same.

2.6 User characteristics

ADMIN

Admin has the full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access to the system.

Key functions:

•Access patient record, doctor Record.

•Add new doctor entry in system database.

• Confirm Payment and Generate Bill.

- View Records.(Total no of patients treated, doctor added/remove, consultant fee).

2.7 Constraints

- System is wirelessly networked with an encryption.
- System is only accessible within the hospital's website only.
- Database is password protected.
- Should use less RAM and processing power.
- Each user should have individual ID and password.
- Only administrator can access the whole system.

CHAPTER 3

SPECIFIC REQUIREMENTS

3.1 PERFORMANCE REQUIREMENTS

- **Response time**- The system will give responses within 1 second after checking the patient information and other information.
- **Capacity**-The system must support 1000 people at a time
- **User interface**- User interface screen will response within 5 seconds

3.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure. All the administrative and data entry operators have unique logins so system can understand who is login in to system right now no intruders allowed except system administrative nobody cannot change record and valuable data.

3.3 SECURITY REQUIREMENTS

1. Want take the responsibility of failures due to hardware malfunctioning.
2. Warranty period of maintaining the software would be one year.
3. Additional payments will be analyzed and charged for further maintenance.
4. If any error occur due to a user's improper use. Warranty will not be allocated to it.
5. No money back returns for the software.

3.4 SOFTWARE SYSTEM ATTRIBUTES

3.4.1 Usability: Software can be used again and again without distortion.

3.4.2 Availability: The system shall be available all the time.

3.4.3 Correctness: Bug free software which fulfills the correct need/requirements of the client.

3.4.4 Maintainability: The ability to maintain, modify information and update fix problems of the system.

3.4.5 <u>Accessibility</u>: Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.

3.5 FUNCTIONAL REQUIREMENTS

S.No.	MODULE NAME	APPLICABLE ROLES	DESCRIPTION
1.	REGISTRATION	PATIENT	**PATIENT:** Can Register by filling all the required details, after this the system will verify the details and check if already registered or not.
2.	MAKE APPT.	PATIENT	**PATIENT:** Can Select doctor, date time and make an appointment request after this system shall show a confirmation for appointment request.
3.	CANCL APPT.	PATIENT DOCTOR	**PATIENT :** Can Cancel appointment if want to by just one click after this system shall ask for re-schedule or refund of payment. **DOCTOR :** Can Cancel appointment if want to by just one click after this system shall send a message to the patient.
4.	DOCTOR MODULE	ADMIN	**ADMIN :** Can add a new doctor by filling all the details after this system shall show a confirmation message. Can Remove a doctor by just one click after this system shall show confirmation message.

CHAPTER 4

DESIGN

4.1 DATA DICTIONARY

1.	legal_character	[a-z| A-Z]
2.	Dig it	[0-9]
3.	special_ch	[@|$|#|+|-]
4.	Blood	[A|B|AB|O]

1.	Name	first_name + (middle_name) + last_name
2.	first_name	{legal_character}*
3.	middle_name	{legal_character}*
4.	last_name	{legal_character}*
5.	P_ID	{legal_character + digit}*
6.	D_ID	{legal_character + digit}*
7.	A_ID	{legal_character + digit}*
8.	Password	{legal_character + digit + special_ch}*
9.	Address	House_no + (Street) + City
10.	City	{legal_character}*
11.	Mobile No.	{ digit }*
12.	Blood_Group	{Blood + special_ch}*

Table 4.1 Data Dictionary

4.2 ER DIAGRAM

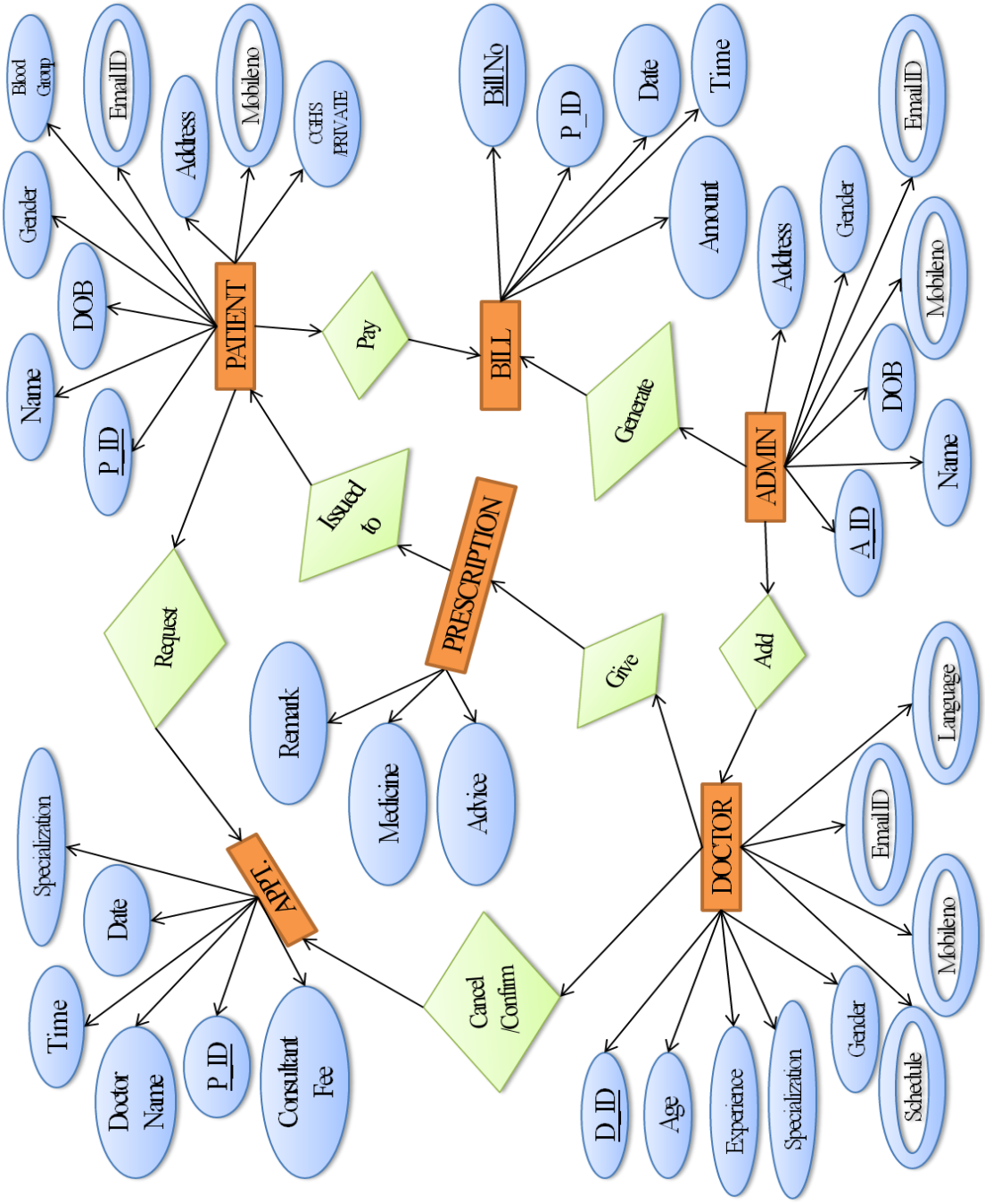

FIGURE 4.1 ER Diagram

4.3 DATA DESIGN

S NO.	COLUMN NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	P_ID	Varchar(50)	Primary Key	Contains Unique Id
2.	Name	Varchar(50)	-	Contains Name
3.	DOB	Varchar(50)	-	Contains Date Of Birth
4.	Gender	Varchar(50)	-	Contains Gender
5.	Blood Group	Varchar(50)	-	Contains Blood Group
6.	Email ID	Varchar(50)	-	Contains Email Id
7.	Address	Varchar(50)	-	Contains Address
8.	Mobile No.	Integer	-	Contains Mobile No.
9.	CGHS/Private	Varchar(50)	-	Contains Category

Table 4.2 Patient

S NO.	COLUMN NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	P_ID	Varchar(50)	Primary Key	Contains Unique Id Patient
2.	Specialization	Varchar(50)	-	Contains Name of the Department in which Patient wants to visit
3.	Doctor's Name	Varchar(50)	-	Contains Doctor Name Patient Wants To Visit
4.	Date	Date	-	Contains Date For The Appointment

Table 4.3 Appointment

S NO.	COLUMN NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	D_ID	Varchar(50)	Primary Key	Contains unique ID
2.	Age	Integer	-	Contains age
3.	Gender	Varchar(50)	-	Contains gender
4.	Specialization	Varchar(50)	-	Contains specialization
5.	Experience	Varchar(50)	-	Contains experience of the doctor (In months)
6.	Language	Varchar(50)	-	Contains in how many languages doctor can speak.
7.	Mobile No.	Integer	-	Contains mobile number
8.	Email ID	Varchar(50)	-	Contains Email Id
9.	Schedule	Varchar(50)	-	Contains day and time for which the doctor is available

Table 4.4 Doctor

S NO.	COLUMN NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	D_ID	Varchar(50)	-	Contains unique ID
2.	P_ID	Varchar(50)	Primary Key	Contains unique ID
3.	Medicine	Varchar(50)		Contains name of the medicine.
4.	Remark	Varchar(50)		Contains Remark given by the doctor for the patient.

Table 4.5 Prescription

S NO.	COLUMN NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	A_ID	Varchar(50)	Primary Key	Contains unique ID.
2.	Name	Varchar(50)	-	Contains Name
3.	DOB	Varchar(50)	-	Contains Date Of Birth
4.	Gender	Varchar(50)	-	Contains Gender
5.	Email ID	Varchar(50)	-	Contains Email Id
6.	Mobile No.	Integer	-	Contains Mobile No.
7.	Address	Varchar(50)	-	Contains Address

Table 4.6 Admin

4.4 COMPONENT LEVEL DIAGRAM

Table Create

```
cur.execute("create database if not exists hello")

cur.execute("use hello")

cur.execute("create table if not exists apt"
        "("
        "idno varchar(12) primary key,"
        "name char(20),"
        "age char(3),"
        "gender char(1),"
        "phone varchar(10),"
        "bg varchar(3))")
```

Registration Module

```
def register():
```

```
    global e1,e2,e3,e4,e5,e6
    root1=Tk()



    label=Label(root1,text="REGISTER YOURSELF",font='arial 25 bold')
    label.pack()
    frame=Frame(root1,height=500,width=200)
    frame.pack()
    l1=Label(root1,text="PATIENT ID")
    l1.place(x=10,y=130)
    e1=tkinter.Entry(root1)
    e1.place(x=100,y=130)
    l2=Label(root1,text="NAME")
    l2.place(x=10,y=170)
    e2=tkinter.Entry(root1)
    e2.place(x=100,y=170)
    l3=Label(root1,text="AGE")
    l3.place(x=10,y=210)
    e3=tkinter.Entry(root1)
    e3.place(x=100,y=210)
    l4=Label(root1,text="GENDER M\F")
    l4.place(x=10,y=250)
    e4=tkinter.Entry(root1)
    e4.place(x=100,y=250)
    l5=Label(root1,text="PHONE")
    l5.place(x=10,y=290)
    e5=tkinter.Entry(root1)
    e5.place(x=100,y=290)
    l6=Label(root1,text="BLOOD GROUP")
    l6.place(x=10,y=330)
    e6=tkinter.Entry(root1)
    e6.place(x=100,y=330)
    b1=Button(root1,text="SUBMIT",command=entry)
    b1.place(x=150,y=370)

    root.resizable(False,False)
    root1.mainloop()
```

Message for Appointment

```
def apo_details():
```

```
    global x2
    p1=x2.get()
    if int(p1)==1:
        i=("Dr. Varun \nRoom no:- 201")
        j=("Dr. Hrithik \nRoom no:- 202")
        q=(i,j)
        h=rd.choice(q)
        u=(23,34,12,67,53,72)
        o=rd.choice(u)
        det=("Your appointment is fixed with",h,"\nDate:-",datetime.date.today() +
datetime.timedelta(days=3),'\nAppointmnet no:-',o)
        tkinter.messagebox.showinfo("APPOINTMENT DETAILS",det)

    elif int(p1)==2:
        i=("Dr. Sidharth \nRoom no. 207")
        j=("Dr. Abhishek \nRoom no. 208")
        q=(i,j)
        h=rd.choice(q)
        u=(23,34,12,67,53,72)
        o=rd.choice(u)
        det=("Your appointment is fixed with",h,"\nDate:-",datetime.date.today() +
datetime.timedelta(days=5),'\nAppointmnet no:-',o)
        tkinter.messagebox.showinfo("APPOINTMENT DETAILS",det)

    elif int(p1)==3:
        i=("Dr. Salman \nRoom no. 203")
        j=("Dr. Shahrukh \nRoom no. 204")
        q=(i,j)
        h=rd.choice(q)
        u=(23,34,12,67,53,72)
        o=rd.choice(u)
        det=("Your appointment is fixed with",h,"\nDate:-",datetime.date.today() +
datetime.timedelta(days=3),'\nAppointmnet no:-',o)
        tkinter.messagebox.showinfo("APPOINTMENT DETAILS",det)

    elif int(p1)==4:
        i=("Dr. Ajay, \nRoom no. 209")
        j=("Dr. Ranveer \nRoom no. 200")
        q=(i,j)
        h=rd.choice(q)
        u=(23,34,12,67,53,72)
```

```
        o=rd.choice(u)
        det=("Your appointment is fixed with",h,"\nDate:-",datetime.date.today() +
datetime.timedelta(days=6),'\nAppointmnet no:-',o)
        tkinter.messagebox.showinfo("APPOINTMENT DETAILS",det)

    elif int(p1)==5:
        i=("Dr. Akshay \nRoom no. 205")
        j=("Dr. Amir \nRoom no. 206")
        q=(i,j)
        h=rd.choice(q)
        u=(23,34,12,67,53,72)
        o=rd.choice(u)
        det=("Your appointment is fixed with",h,"\nDate:-",datetime.date.today() +
datetime.timedelta(days=4),'\nAppointmnet no:-',o)
        tkinter.messagebox.showinfo("APPOINTMENT DETAILS",det)

    elif int(p1)==6:
        i=("Dr. Irfan \nRoom no. 001")
        j=("Dr. John \nRoom no. 002")
        k=("Dr. Sanjay \nRoom no. 003")
        l=("Dr. Shahid \nRoom no. 004")
        q=(i,j,k,l)
        h=rd.choice(q)
        u=(23,34,12,67,53,72)
        o=rd.choice(u)
        det=("Your appointment is fixed with",h,"\nDate:-",datetime.date.today() +
datetime.timedelta(days=1),'\nAppointmnet no:-',o)
        tkinter.messagebox.showinfo("APPOINTMENT DETAILS",det)

    else:
        tkinter.messagebox.showwarning('WRONG INPUT','PLEASE ENTER VALID
VALUE')
```

For Appointment

```
def get_apoint():
   global x1,x2
   p1=x1.get()
   cur.execute('select * from apt where idno=(%s)',(p1,))
   dat=cur.fetchall()
```

```
a=[]
for i in dat:
   a.append(i)
if len(a)==0:
   tkinter.messagebox.showwarning("ERROR", "NO DATA FOUND!!")
else:
   root3=Tk()
   label=Label(root3,text="APPOINTMENT",font='arial 25 bold')
   label.pack()
   frame=Frame(root3,height=500,width=300)
   frame.pack()
   if i[3]=='M' or i[3]=='m':
         x="Mr."
         name2=Label(root3,text=i[1])
         name2.place(x=140,y=80)
   else:
         x="Mrs\Ms."
         name2=Label(root3,text=i[1])
         name2.place(x=170,y=80)
   for i in dat:
      name=Label(root3,text='WELCOME')
      name.place(x=50,y=80)
      name1=Label(root3,text=x)
      name1.place(x=120,y=80)
      age=Label(root3,text='AGE:-')
      age.place(x=50,y=100)
      age1=Label(root3,text=i[2])
      age1.place(x=100,y=100)
      phone=Label(root3,text='PHONE:-')
      phone.place(x=50,y=120)
      phone1=Label(root3,text=i[4])
      phone1.place(x=100,y=120)
      bg=Label(root3,text='BLOOD GROUP:-')
      bg.place(x=50,y=140)
      bg1=Label(root3,text=i[5])
      bg1.place(x=150,y=140)


   L=Label(root3,text='DEPARTMENTS')
   L.place(x=50,y=220)
   L1=Label(root3,text="1.Cardiologist ")
```

```
    L1.place(x=50,y=250)
    L2=Label(root3,text='2.Rheumatologist')
    L2.place(x=50,y=270)
    L3=Label(root3,text='3.Psychitrist')
    L3.place(x=50,y=290)
    L4=Label(root3,text='4.Neurologist')
    L4.place(x=50,y=310)
    L5=Label(root3,text='5.Otolaryngonologist')
    L5.place(x=50,y=330)
    L6=Label(root3,text='6.MI Room')
    L6.place(x=50,y=350)
    L7=Label(root3,text='Enter')
    L7.place(x=100,y=370)
    x2=tkinter.Entry(root3)
    x2.place(x=150,y=370)
    B1=Button(root3,text='Submit',command=apo_details)
    B1.place(x=120,y=440)
    root3.resizable(False,False)
    root3.mainloop()
```

For Adhaar no input

```
def apoint():
   global x1
   root2=Tk()
   label=Label(root2,text="APPOINTMENT",font='arial 25 bold')
   label.pack()
   frame=Frame(root2,height=200,width=200)
   frame.pack()
   l1=Label(root2,text="ADHAAR NO.")
   l1.place(x=10,y=130)
   x1=tkinter.Entry(root2)
   x1.place(x=100,y=130)
   b1=Button(root2,text='Submit',command=get_apoint)
   b1.place(x=100,y=160)
   root2.resizable(False,False)
   root2.mainloop()
```

List of doctors

```
def lst_doc():
    root4=Tk()

    l=["Dr. Varun","Dr. Hrithik","Dr. Salman","Dr. Shahrukh","Dr. Akshay","Dr. Amir","Dr. Sidharth","Dr. Abhishek","Dr. Ajay","Dr. Ranveer",'Dr. Irfan','Dr. John','Dr. Sanjay','Dr. Shahid']

m=["Cardiologist","Cardiologist","Psychitrist","Psychitrist","Otolaryngonologist","Otolaryngonologist","Rheumatologist","Rheumatologist","Neurologist","Neurologist",'MI room','MI room','MI room','MI room']
    n=[201,202,203,204,205,206,207,208,209,200,401,402,403,404]

    frame=Frame(root4,height=500,width=500)
    frame.pack()


    l1=Label(root4,text='NAME OF DOCTORS')
    l1.place(x=20,y=10)
    count=20
    for i in l:
        count=count+20
        l=Label(root4,text=i)
        l.place(x=20,y=count)

    l2=Label(root4,text='DEPARTMENT')
    l2.place(x=140,y=10)
    count1=20
    for i in m:
        count1=count1+20
        l3=Label(root4,text=i)
        l3.place(x=140,y=count1)

    l4=Label(root4,text='ROOM NO')
    l4.place(x=260,y=10)
    count2=20
    for i in n:
        count2=count2+20
        l5=Label(root4,text=i)
        l5.place(x=260,y=count2)
    root.resizable(False,False)
```

```
    root4.mainloop()
```

Service Available

```
def ser_avail():
    root5=Tk()
    frame=Frame(root5,height=500,width=500)
    frame.pack()
    l1=Label(root5,text='SERVICES AVAILABLE')
    l1.place(x=20,y=10)
    f=["X-Ray","MRI","CT                    Scan","Endoscopy","Dialysis","Ultrasound
","EEG","ENMG","ECG"]
    count1=20
    for i in f:
        count1=count1+20
        l3=Label(root5,text=i)
        l3.place(x=20,y=count1)
    l2=Label(root5,text='ROOM NO.')
    l2.place(x=140,y=10)
    g=[101,102,103,104,105,301,302,303,304]
    count2=20
    for i in g:
        count2=count2+20
        l4=Label(root5,text=i)
        l4.place(x=140,y=count2)
    l5=Label(root5,text='To avail any of these please contact on our no.:- 8630877158,
8429169872')
    l5.place(x=20,y=240)
    root5.resizable(False,False)
    root5.mainloop()
```

Modification

```
def mod_sub():
    global x3
    root7=Tk()
    label=Label(root7,text="MODIFICATION",font='arial 25 bold')
    label.pack()
```

```
    frame=Frame(root7,height=200,width=200)
    frame.pack()
    l1=Label(root7,text="ADHAAR NO.")
    l1.place(x=10,y=130)
    x3=tkinter.Entry(root7)
    x3.place(x=100,y=130)
    b1=Button(root7,text='Submit',command=modify)
    b1.place(x=100,y=160)
    root7.resizable(False,False)
    root7.mainloop()



def modify():
    global x3,x4
    p1=x3.get()
    cur.execute('select * from apt where idno=(%s)',(p1,))
    dat=cur.fetchall()
    a=[]
    for i in dat:
        a.append(i)
    if len(a)==0:
        tkinter.messagebox.showwarning("ERROR", "NO DATA FOUND!!")
    else:
      root6=Tk()
      frame=Frame(root6,height=500,width=500)
      frame.pack()
      l1=Label(root6,text='DATA MODIFICATION',font="arial 15 bold")
      l1.place(x=75,y=10)
      l2=Label(root6,text='WHAT YOU WANT TO CHANGE')
      l2.place(x=50,y=200)
      l3=Label(root6,text='1.NAME')
      l3.place(x=50,y=220)
      l4=Label(root6,text='2.AGE')
      l4.place(x=50,y=240)
      l5=Label(root6,text='3.GENDER')
      l5.place(x=50,y=260)
      l6=Label(root6,text='4.PHONE')
      l6.place(x=50,y=280)
      l7=Label(root6,text='5.BLOOD GROUP')
```

```
    l7.place(x=50,y=300)
    x2=Label(root6,text='Enter')
    x2.place(x=50,y=330)
    x4=tkinter.Entry(root6)
    x4.place(x=100,y=330)
    for i in dat:
        name=Label(root6,text='NAME:-')
        name.place(x=50,y=80)
        name1=Label(root6,text=i[1])
        name1.place(x=150,y=80)
        age=Label(root6,text='AGE:-')
        age.place(x=50,y=100)
        age1=Label(root6,text=i[2])
        age1.place(x=150,y=100)
        gen=Label(root6,text='GENDER:-')
        gen.place(x=50,y=120)
        gen1=Label(root6,text=i[3])
        gen1.place(x=150,y=120)
        pho=Label(root6,text='PHONE:-')
        pho.place(x=50,y=140)
        pho1=Label(root6,text=i[4])
        pho1.place(x=150,y=140)
        bg=Label(root6,text='BLOOD GROUP:-')
        bg.place(x=50,y=160)
        bg1=Label(root6,text=i[4])
        bg1.place(x=150,y=160)
    b=Button(root6,text='Submit')
    b.place(x=50,y=400)
    L1=Label(root6,text='OLD DETAILS')
    L1.place(x=50,y=50)
    L2=Label(root6,text='ENTER NEW DETAIL')
    L2.place(x=50,y=360)
    x5=tkinter.Entry(root6)
    x5.place(x=160,y=360)

    root6.resizable(False,False)
    root6.mainloop()
```

Body

```
root=Tk()
label=Label(root,text="CITY HOSPITAL",font="arial 40 bold",bg='blue')
b1=Button(text="Registration",font="arial 20 bold",bg='cyan',command=register)
b2=Button(text="Appointment",font="arial 20 bold",bg='cyan',command=apoint)
b3=Button(text="List of Doctors",font="arial 20 bold",bg='cyan',command=lst_doc)
b4=Button(text="Services available",font='arial 20 bold',bg='cyan',command=ser_avail)
b5=Button(text="Modify data",font='arial 20 bold',bg='cyan',command=mod_sub)
b6=Button(text="Exit",font='arial 20 bold',command=root.destroy,bg='yellow')
label.pack()
b1.pack(side=LEFT,padx=10)
b2.pack(side=LEFT,padx=10)
b3.pack(side=LEFT,padx=10)
b4.pack(side=LEFT,padx=10)
b5.pack(side=LEFT,padx=10)
b6.pack(side=LEFT,padx=10)
frame=Frame(root,height=500,width=100)
frame.pack()
root.resizable(False,False)
root.mainloop()
```

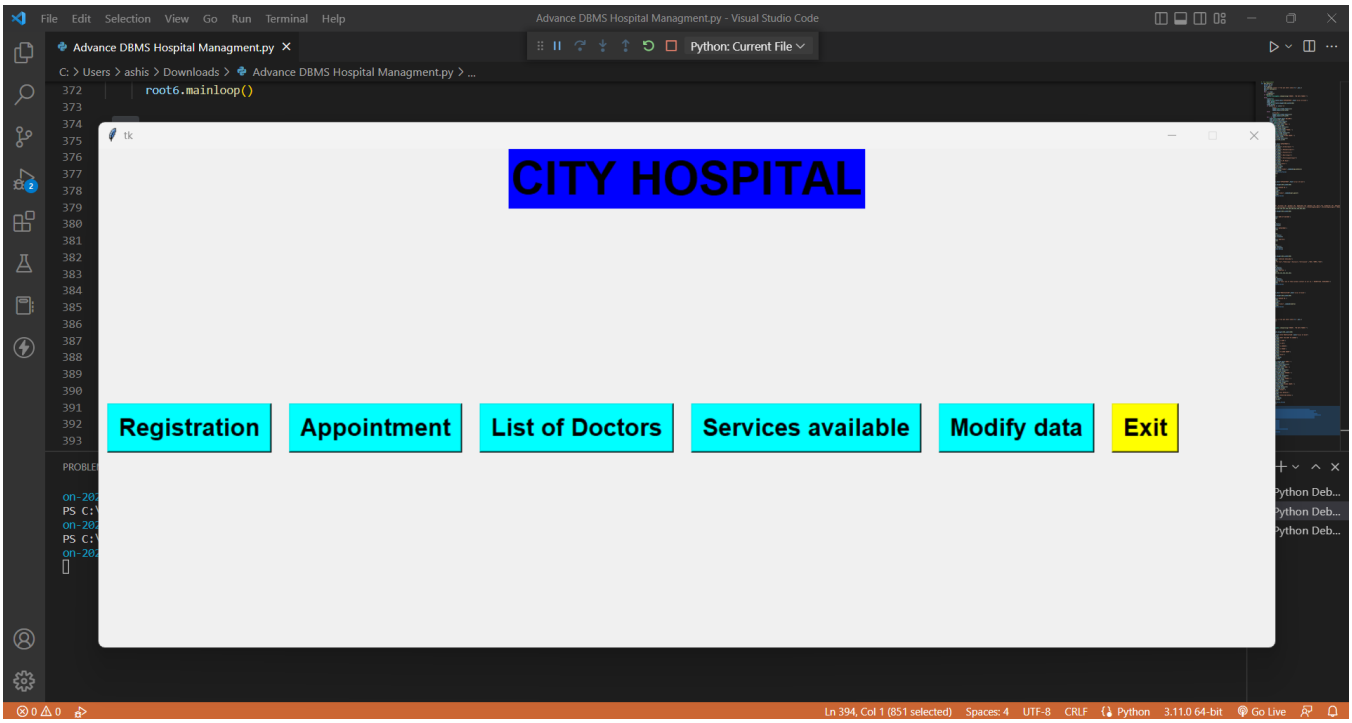

FIGURE 4.2 HOME PAGE

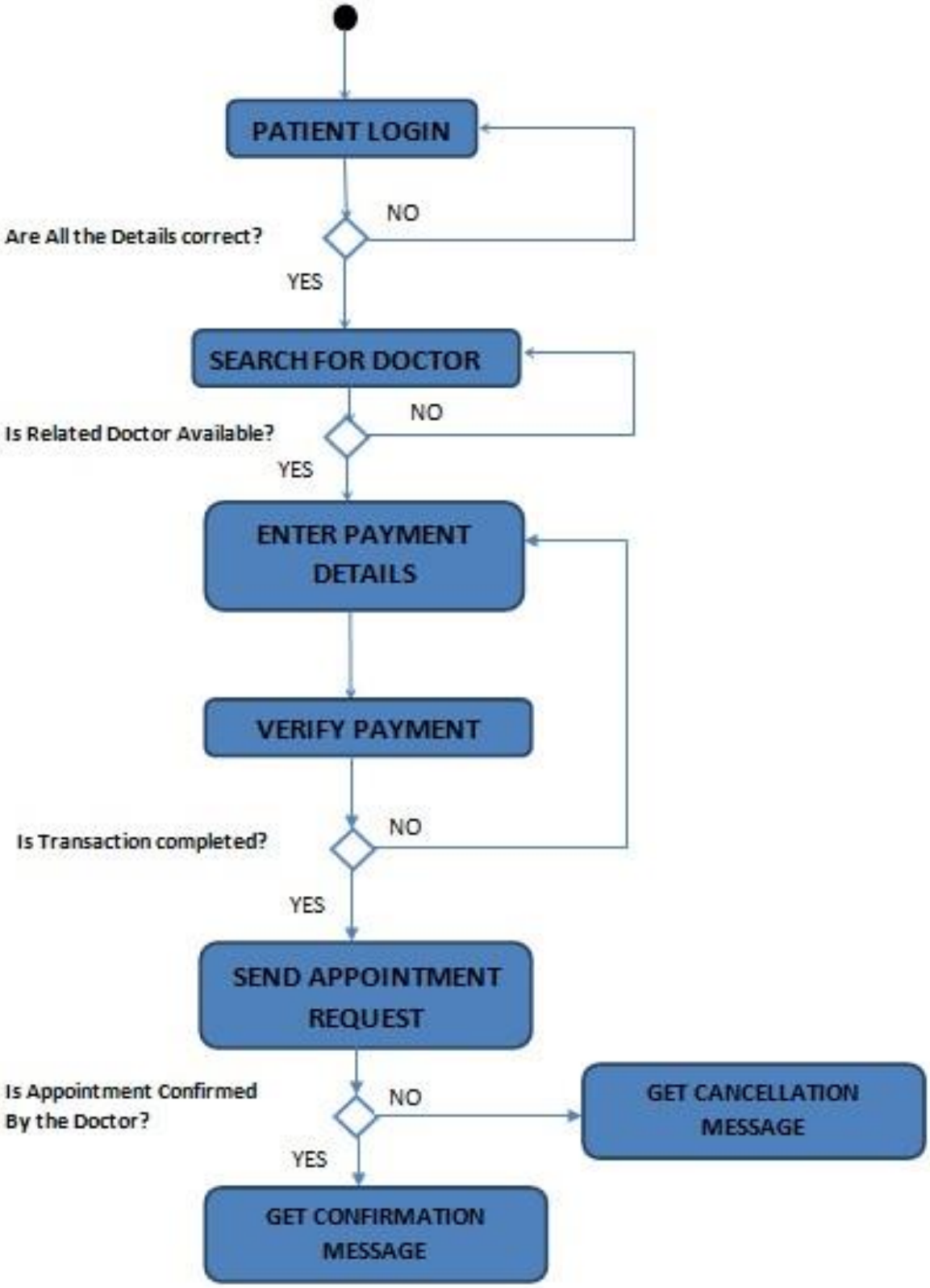

FIGURE 4.3 Component Table Diagram

CHAPTER 5

PROJECT SCREENSHOTS

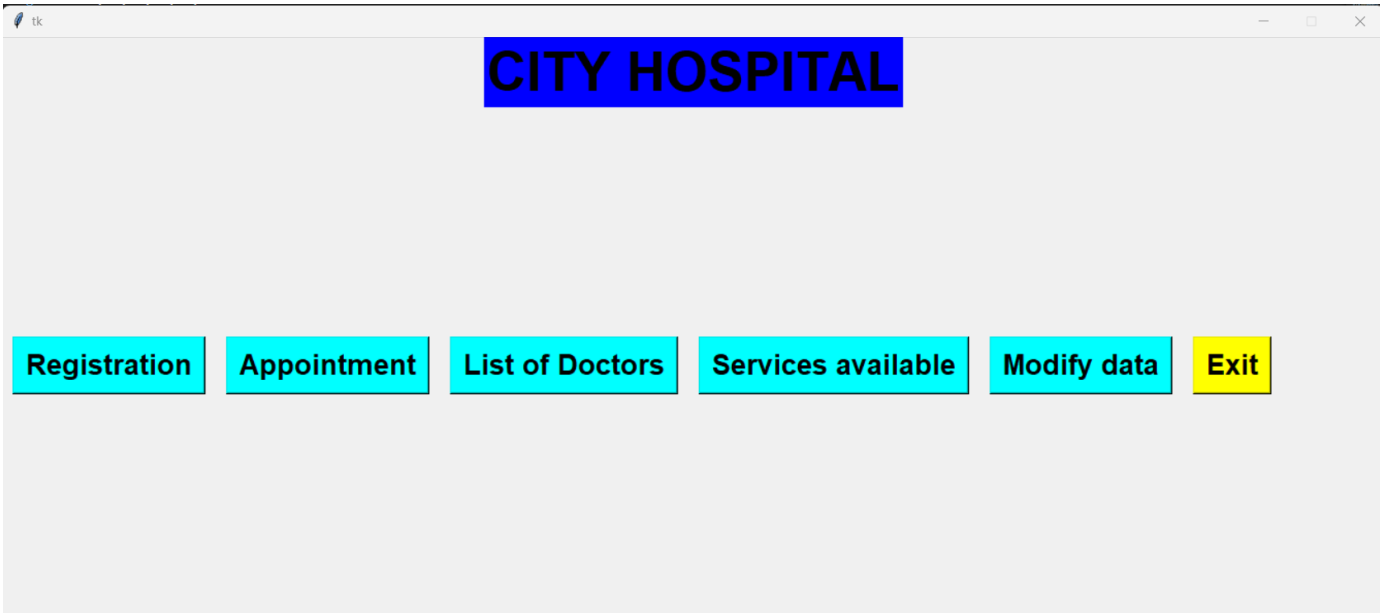

FIGURE 5.1 Home Page

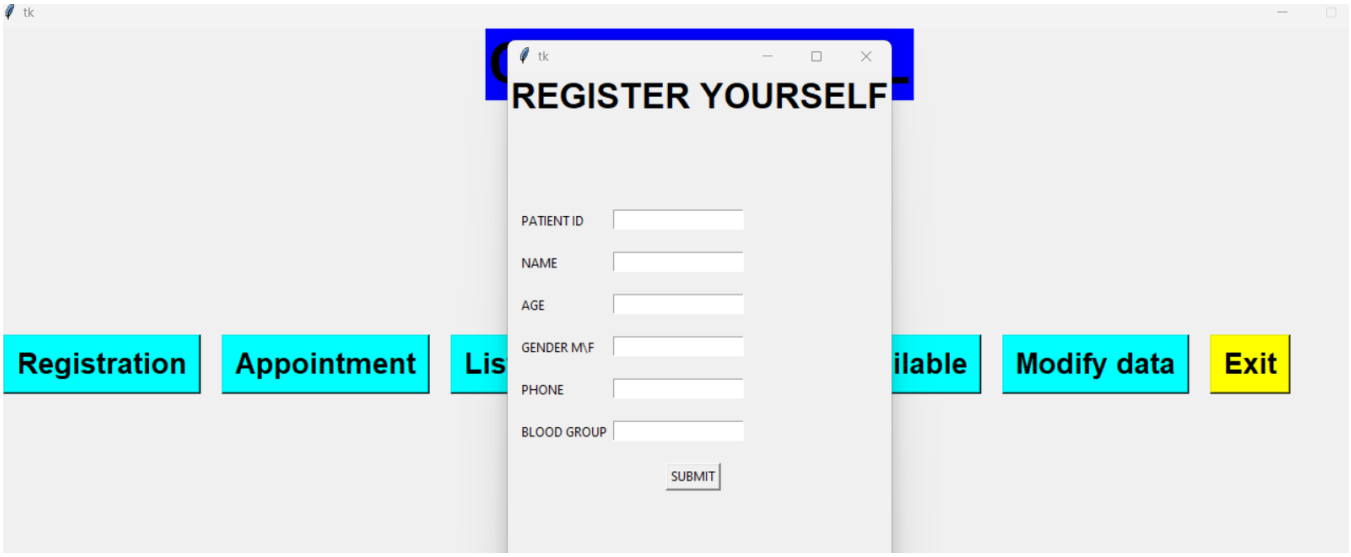

FIGURE 5.2 Register

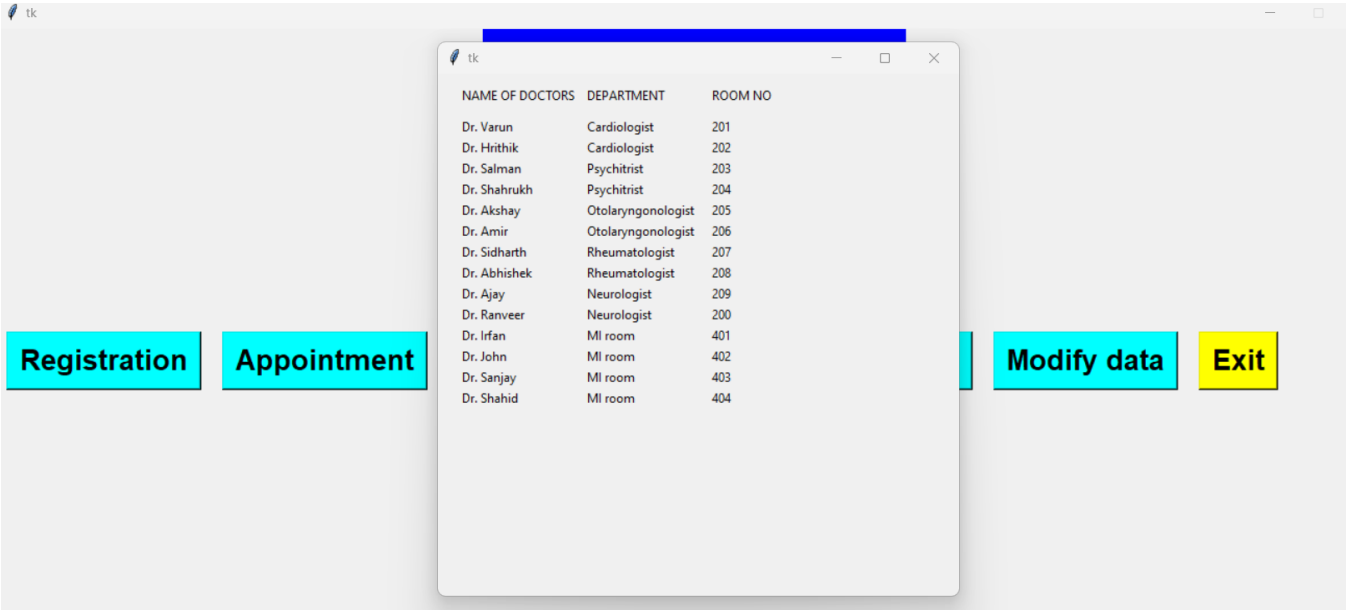

FIGURE 5.3 List Of Doctor

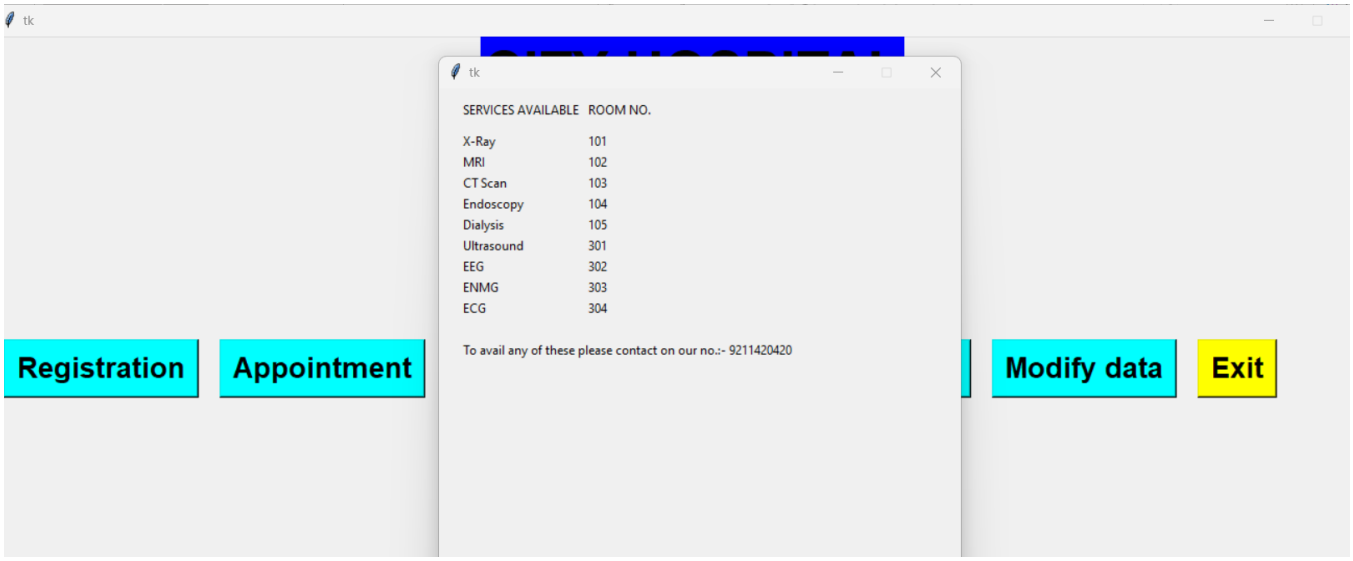

FIGURE 5.4 Service Available

CHAPTER – 6

CONCLUSION

Working on the project was an excellent experience. It helped us to understand the importance of planning, designing and implementation so far we have learnt in our theory books. It helped us unleashing our creativity while working in a team. It also realized the importance of team working, communication as a part of this project.

The project was successfully completed after a lot of efforts and work hours. This project underwent number of compiling, debugging, removing errors, making it bug free, adding more facilities in Hospital Management System and interactivity making it more reliable and useful.

This project focused that scheduling a project and adhering to that schedule creates a hard sense of time- management. It has also let us known that co-operative teamwork always produce effective results.

The entire project has been developed and deployed as per the requirements stated by the user. It is found to be bug free as per the testing standards that are implemented.

The estimated cost of the project is (efforts) 12 and the estimated size of the project is (FP) 209.72.

There are also few features which can be integrated with this system to make it more flexible. Below list shows the future points to be consider :

- Getting the current status of patient.
- Including a different module for pharmacy, LAB, Bed Allotment and many more.
- Including a Frequently Asked Questions Section.

Finally, we like to conclude that we put all our efforts throughout the development of our project and tried to fulfill most of the requirements of the user.

REFERENCE

- https://www.python.com/
- https://wwwa.mysql.com/
- https://www.google.com/
- https://www.geeksforgeeks