Shaheed Zulfikar Ali Bhutto
Institute of Science & Technology

“PROJECT REPORT”

COURSE: DATA STRUCTURES AND ALGORITHMS

Group Members:

Roha Ali, 2112124

Syeda Mahnoor Hasan, 2112135

Asma Hafiz Mir Nawaz, 2112106

Hussam Wasti, 2112113

Respected Teachers:

Syed Muhammad Hassan (Lab)

Muhammad Talha Javaid (Theory)

December/2022

Project:

"HOSPITAL MANAGEMENT SYSTEM"

Index

Contents: Page No

Summary:

The purpose of the Hospital Management System project is to develop computerized management system software, which is user friendly, simple, fast and cost effective. It deals with

the collection of patient’s information like adding patient, update patient, search patient by ID and Contact, view patient and same operations are performed for Doctor Information. There is also a checkup list which according to the priority of the patient assigns a doctor.

Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve these details as and when required, and to manipulate these details meaningfully. In this system we have used two data structures which are a doubly linked list and priority queue. The records which are stored in the system are well protected for personal use and make the data process very fast.

Problem Statement:

It is very important to maintain efficient software to handle information about a hospital. This application provides a way to record this information and to access these in a simple way.

Features:

1. Inserting Patients.
2. Inserting Doctors.
3. Reading information of patients.
4. Reading information about doctors.
5. Assigning Patients to specific doctors.
6. Giving patients priority.
7. Giving Recommendations to the patients.

Limitations:

1. Patients' checkup details can't be viewed at any time.

Code Description:

We are developing a hospital management system. We will use liked lists, priority queues, and file handling. We will use three classes of patients: patients, doctors, and checkups. The patients class contains the patient's id, name, and contact information. The doctor class also contains the doctor's name, specialty, and the fee that the doctor will charge. Finally, we will assign the doctor to the connected patient or the patient who requests to see the doctor.

The private attributes are declared to the patient and the getter setters are created if any new doctors seek to register with the hospital. The same is true for patients; as new patients visit the hospital frequently; the attributes are privately declared to the patient.

The to-string is also provided for the patient and doctor to present all information in the main menu's output. The checkup list is created with the doctor and patient classes formed as inheritance, the priority is made, and that is the patient wants to meet in an emergency or the patient wants to make a regular visit. Finally, in the checkup, the doctor's advice to the patient about the diet or any other take care the patient wants to follow is made to the checkup list. In other words, the checkup list reflects the doctor and patient qualities.

In a newly created class of patient list, a node called PNODE that has two pointers labelled previous and next was created. Patient has one function called insert, which is used to add patients from the class of patients that contains their attributes. The patient list employs a doubly linked list. Using the attributes, we may create functions that have the same names as the patient attributes, such as searching by contact information, name, or patient ID. The function with the name size is where the number of patients—both those present and those wanting to see a doctor—will be printed. The last function in the patient list is to print data in which all the data of the patient like the id, name, contact will get the doctor so that they will choose which patient he wants to see next.

With the use of a linked list, we create a node for the doctor list class for the insertion of new physicians, much to how we made the class of the patient list inherited to the patient. In order to search for and update doctors, we make the new functions match their attributes. We have the ability to print all the information about the doctors we entered. We can search for the doctor by contact information, by identification, or by specialization. The print data function has been created so that information about the doctors, including their names, IDs, and contact information, is displayed on the output their specialization, as well as the doctor's fee, are displayed on the screen, and another function called "size" is created in which the number of doctors is oriented. All of the doctors' data, as well as all of their attributes, are output, and the getters will assist new doctors in registering. With the aid of a queue, the first doctor will represent the last, and the last doctor will move up to the first.

We create a checkup list with the aid of a priority queue. We create a new node for the checkup list, and as a result of the queue, we create a function for enqueue that uses getters to move the node pointers to the next data. We also create a function for dequeue that assigns a doctor to a patient so that the doctor can make recommendations and complete the remaining checks. Finally, we create a print function in the list class. The algorithm is made in a distinct process from the recommendation function. Recommendation function is made separately in which the algorithm is made in which the doctor will help the patient specifically to the patient.

We create a new object for each class in the main class, and the scanner uses the user's input to determine what to scan.

We will create a menu in which pressing 1 will bring up the doctor's attributes. The user may then enter the necessary patient information, and once all necessary fields have been filled in, the doctor will be successfully inserted. The required information includes the doctor's name, specialty, contact information, and fees.

If the user presses 2 the enter patient info form will open in which the patient information is required. the information the user needs to ill up is the patient’s id, name and contact of the patient

If the user presses 3 all the doctor’s information will be shown how many doctors r registered to the hospital and what specialty, they have. With the help of the contact or name or the id the user can search the doctor and doctor information will be shown

If the user presses 4 all the patient's information will be shown that how many patients r there and wants to see the doctor whether in emergency or the intermediate. We can search the patients with the help of id, name or contact and their functions r already made in the class of patient list

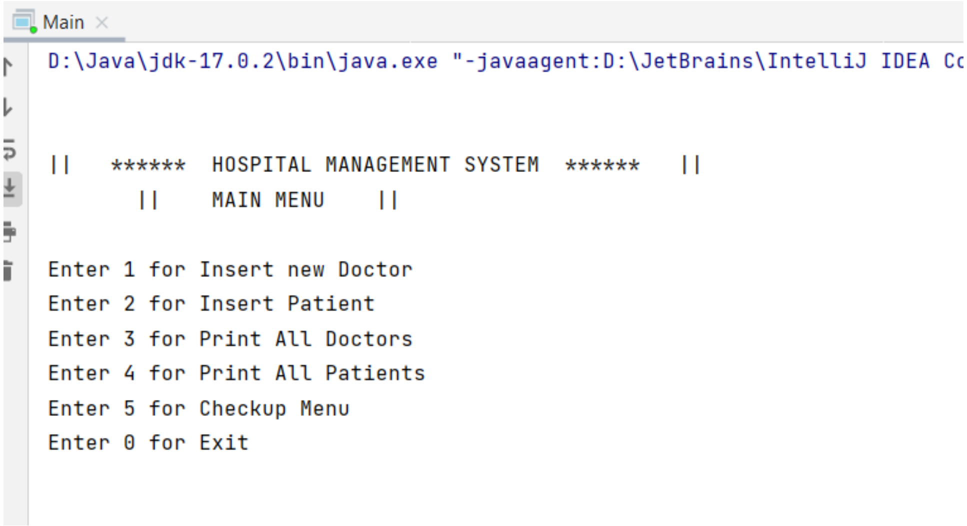
If the user presses 5 the checkup list is shown in which all the doctors and patients information is shown and then the functions of the checkup list class are helpful here, the algorithm in the functions will help the patient that which doctor the patients wants to checkup with and then the statement will appear whether the patient wasn't have emergency or normal checkup.

After the allotting to the doctor, we will ask the user whether to terminate the program or to further allot the patient if any patient Is left if will show separately and the allotted patients will be soon with doctor.

If the user presses 0 the program will stop.

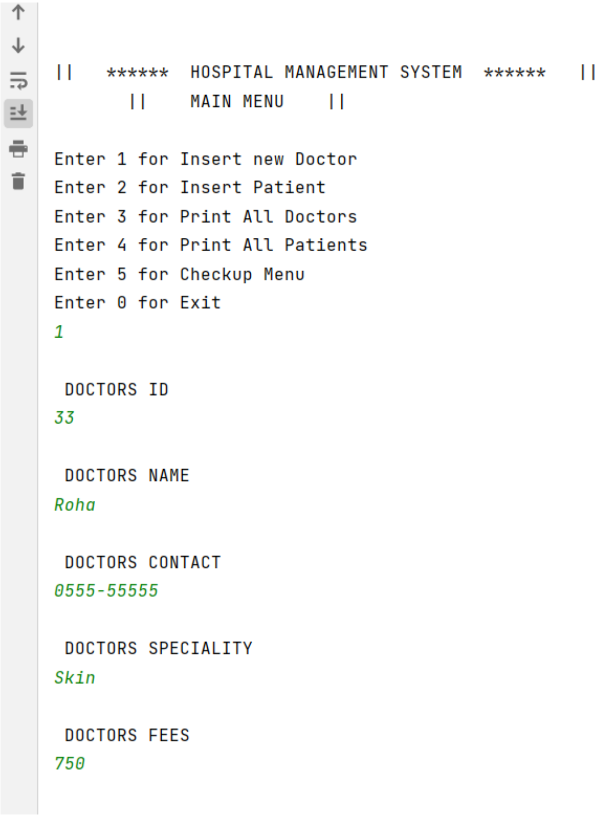
Output:

Main Menu



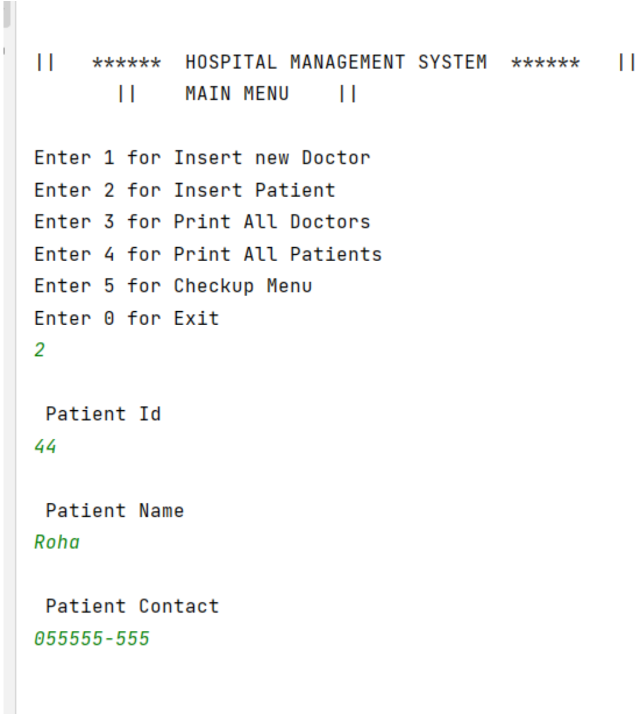
```
D:\Java\jdk-17.0.2\bin\java.exe "-javaagent:D:\JetBrains\IntelliJ IDEA Co


||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
       ||    MAIN MENU    ||

Enter 1 for Insert new Doctor
Enter 2 for Insert Patient
Enter 3 for Print All Doctors
Enter 4 for Print All Patients
Enter 5 for Checkup Menu
Enter 0 for Exit
```

Insert Doctor



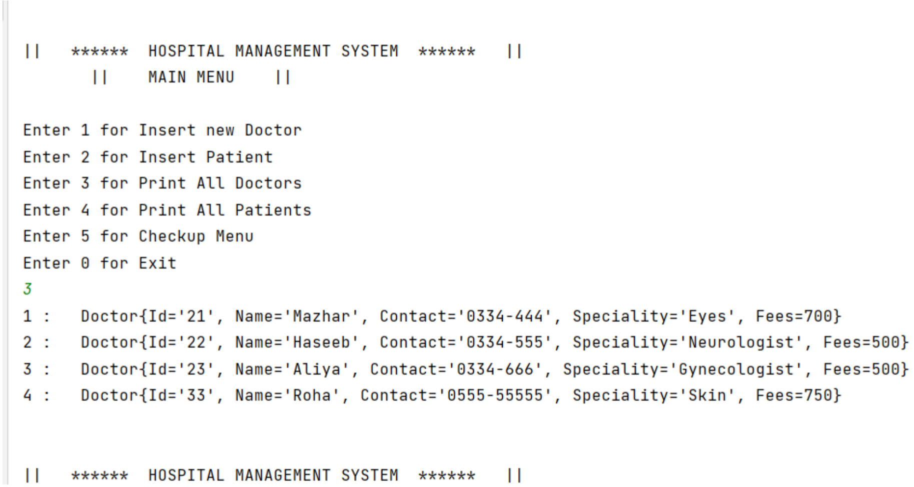
```
||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
       ||    MAIN MENU    ||

Enter 1 for Insert new Doctor
Enter 2 for Insert Patient
Enter 3 for Print All Doctors
Enter 4 for Print All Patients
Enter 5 for Checkup Menu
Enter 0 for Exit
1

 DOCTORS ID
33

 DOCTORS NAME
Roha

 DOCTORS CONTACT
0555-55555

 DOCTORS SPECIALITY
Skin

 DOCTORS FEES
750
```

Insert Patient



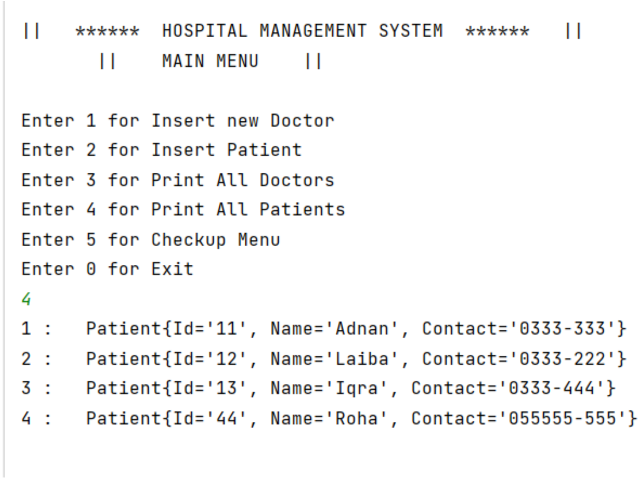
```
||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
       ||    MAIN MENU    ||

Enter 1 for Insert new Doctor
Enter 2 for Insert Patient
Enter 3 for Print All Doctors
Enter 4 for Print All Patients
Enter 5 for Checkup Menu
Enter 0 for Exit
2

 Patient Id
44

 Patient Name
Roha

 Patient Contact
055555-555
```

Print Doctors



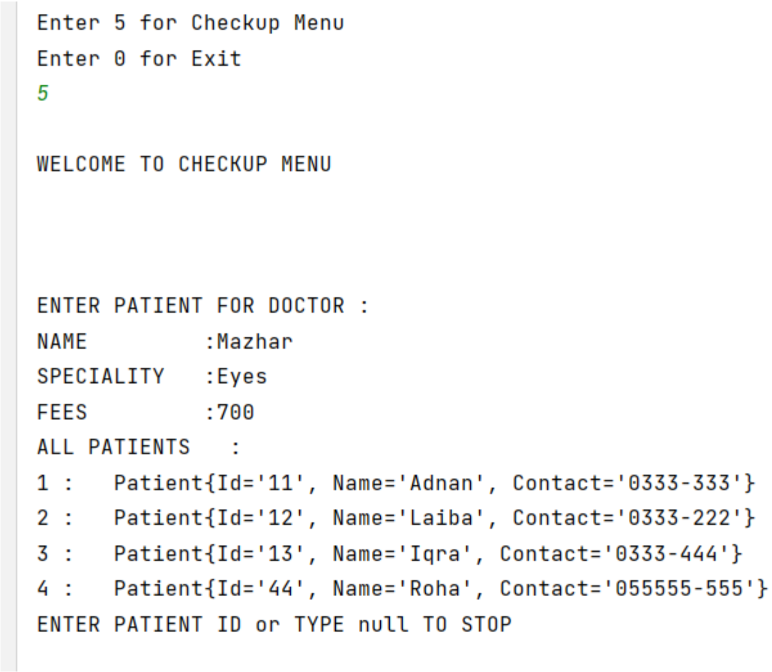
```
||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
       ||    MAIN MENU    ||

Enter 1 for Insert new Doctor
Enter 2 for Insert Patient
Enter 3 for Print All Doctors
Enter 4 for Print All Patients
Enter 5 for Checkup Menu
Enter 0 for Exit
3
1 :   Doctor{Id='21', Name='Mazhar', Contact='0334-444', Speciality='Eyes', Fees=700}
2 :   Doctor{Id='22', Name='Haseeb', Contact='0334-555', Speciality='Neurologist', Fees=500}
3 :   Doctor{Id='23', Name='Aliya', Contact='0334-666', Speciality='Gynecologist', Fees=500}
4 :   Doctor{Id='33', Name='Roha', Contact='0555-55555', Speciality='Skin', Fees=750}


||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
```

Print Patients



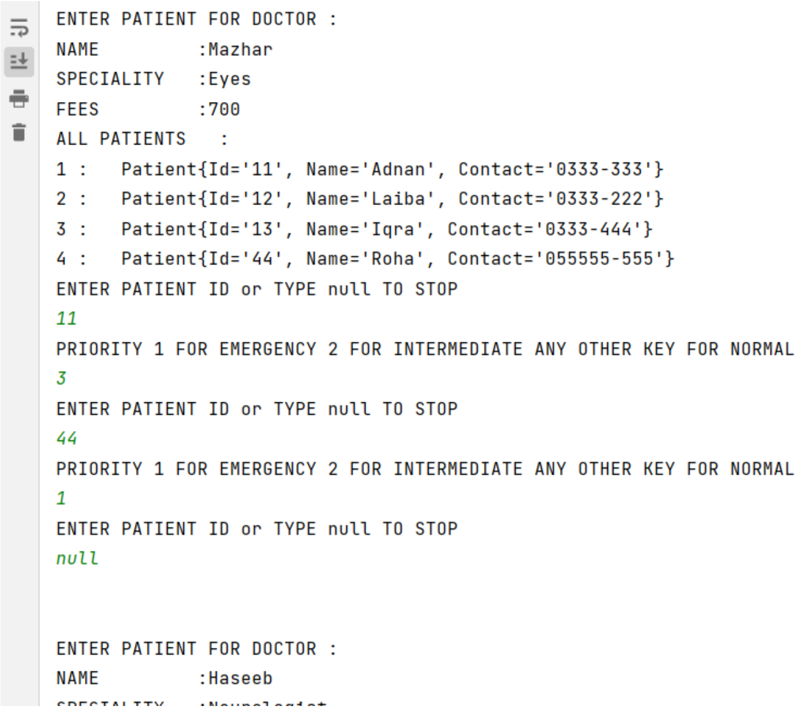
```
||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
       ||    MAIN MENU    ||

Enter 1 for Insert new Doctor
Enter 2 for Insert Patient
Enter 3 for Print All Doctors
Enter 4 for Print All Patients
Enter 5 for Checkup Menu
Enter 0 for Exit
4
1 :   Patient{Id='11', Name='Adnan', Contact='0333-333'}
2 :   Patient{Id='12', Name='Laiba', Contact='0333-222'}
3 :   Patient{Id='13', Name='Iqra', Contact='0333-444'}
4 :   Patient{Id='44', Name='Roha', Contact='055555-555'}
```

Checkup Menu

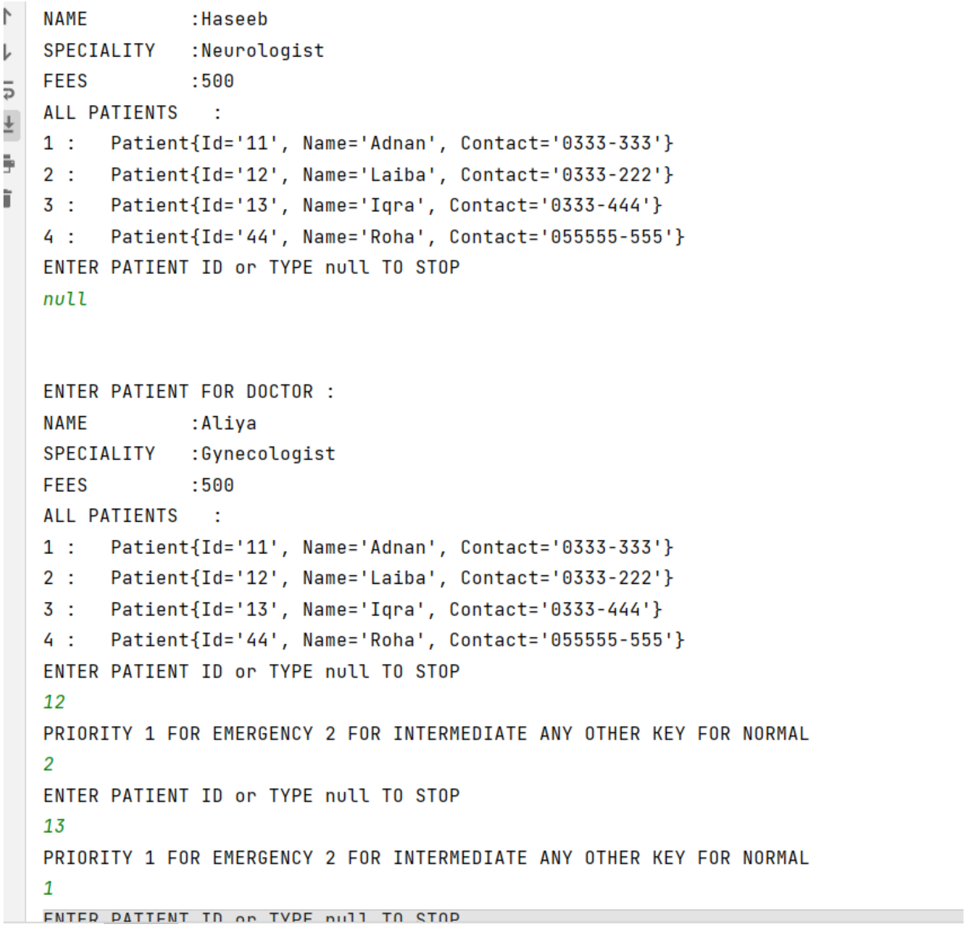


```
Enter 5 for Checkup Menu
Enter 0 for Exit
5

WELCOME TO CHECKUP MENU



ENTER PATIENT FOR DOCTOR :
NAME         :Mazhar
SPECIALITY   :Eyes
FEES         :700
ALL PATIENTS   :
1 :   Patient{Id='11', Name='Adnan', Contact='0333-333'}
2 :   Patient{Id='12', Name='Laiba', Contact='0333-222'}
3 :   Patient{Id='13', Name='Iqra', Contact='0333-444'}
4 :   Patient{Id='44', Name='Roha', Contact='055555-555'}
ENTER PATIENT ID or TYPE null TO STOP
```

Assigning Patients to specific Doctors

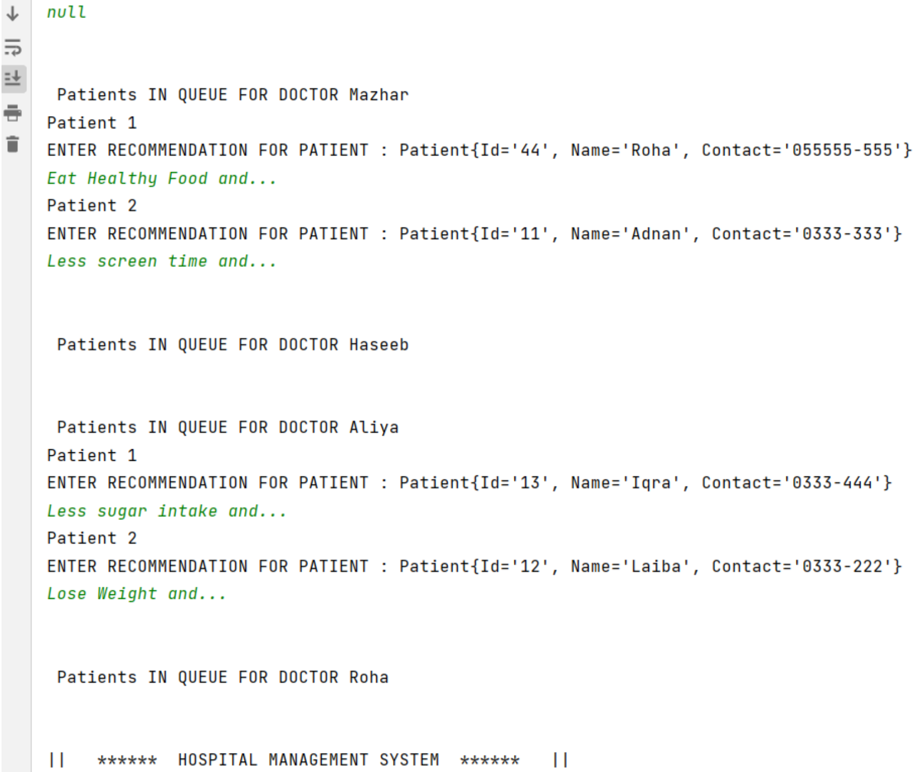


```
ENTER PATIENT FOR DOCTOR :
NAME         :Mazhar
SPECIALITY   :Eyes
FEES         :700
ALL PATIENTS   :
1 :   Patient{Id='11', Name='Adnan', Contact='0333-333'}
2 :   Patient{Id='12', Name='Laiba', Contact='0333-222'}
3 :   Patient{Id='13', Name='Iqra', Contact='0333-444'}
4 :   Patient{Id='44', Name='Roha', Contact='055555-555'}
ENTER PATIENT ID or TYPE null TO STOP
11
PRIORITY 1 FOR EMERGENCY 2 FOR INTERMEDIATE ANY OTHER KEY FOR NORMAL
3
ENTER PATIENT ID or TYPE null TO STOP
44
PRIORITY 1 FOR EMERGENCY 2 FOR INTERMEDIATE ANY OTHER KEY FOR NORMAL
1
ENTER PATIENT ID or TYPE null TO STOP
null


ENTER PATIENT FOR DOCTOR :
NAME         :Haseeb
SPECIALITY   :Neurologist
```



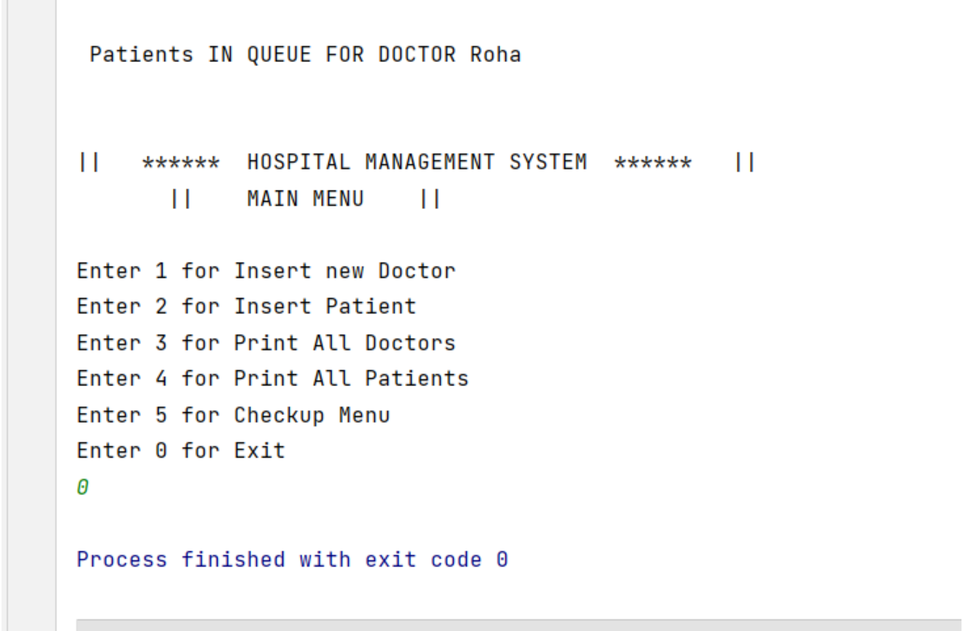
```
NAME          :Haseeb
SPECIALITY    :Neurologist
FEES          :500
ALL PATIENTS    :
1 :   Patient{Id='11', Name='Adnan', Contact='0333-333'}
2 :   Patient{Id='12', Name='Laiba', Contact='0333-222'}
3 :   Patient{Id='13', Name='Iqra', Contact='0333-444'}
4 :   Patient{Id='44', Name='Roha', Contact='055555-555'}
ENTER PATIENT ID or TYPE null TO STOP
null


ENTER PATIENT FOR DOCTOR :
NAME          :Aliya
SPECIALITY    :Gynecologist
FEES          :500
ALL PATIENTS    :
1 :   Patient{Id='11', Name='Adnan', Contact='0333-333'}
2 :   Patient{Id='12', Name='Laiba', Contact='0333-222'}
3 :   Patient{Id='13', Name='Iqra', Contact='0333-444'}
4 :   Patient{Id='44', Name='Roha', Contact='055555-555'}
ENTER PATIENT ID or TYPE null TO STOP
12
PRIORITY 1 FOR EMERGENCY 2 FOR INTERMEDIATE ANY OTHER KEY FOR NORMAL
2
ENTER PATIENT ID or TYPE null TO STOP
13
PRIORITY 1 FOR EMERGENCY 2 FOR INTERMEDIATE ANY OTHER KEY FOR NORMAL
1
ENTER PATIENT ID or TYPE null TO STOP
```

Patients coming in queue for checkup according to their priority



```
null


 Patients IN QUEUE FOR DOCTOR Mazhar
Patient 1
ENTER RECOMMENDATION FOR PATIENT : Patient{Id='44', Name='Roha', Contact='055555-555'}
Eat Healthy Food and...
Patient 2
ENTER RECOMMENDATION FOR PATIENT : Patient{Id='11', Name='Adnan', Contact='0333-333'}
Less screen time and...


 Patients IN QUEUE FOR DOCTOR Haseeb


 Patients IN QUEUE FOR DOCTOR Aliya
Patient 1
ENTER RECOMMENDATION FOR PATIENT : Patient{Id='13', Name='Iqra', Contact='0333-444'}
Less sugar intake and...
Patient 2
ENTER RECOMMENDATION FOR PATIENT : Patient{Id='12', Name='Laiba', Contact='0333-222'}
Lose Weight and...


 Patients IN QUEUE FOR DOCTOR Roha


||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
```

End program



```
 Patients IN QUEUE FOR DOCTOR Roha


||   ******  HOSPITAL MANAGEMENT SYSTEM  ******   ||
       ||    MAIN MENU    ||

Enter 1 for Insert new Doctor
Enter 2 for Insert Patient
Enter 3 for Print All Doctors
Enter 4 for Print All Patients
Enter 5 for Checkup Menu
Enter 0 for Exit
0

Process finished with exit code 0
```

Conclusion:

Using this application, we can retrieve a patient's and doctor's history with a single click. Thus, processing information will be faster. It guarantees accurate maintenance of patient and doctor details and the checkup class. Since we are entering details of the patients and doctors electronically in the Hospital Management System the data will be secured. It easily reduces the bookkeeping task and thus reduces the human effort and increases accuracy speed. The Hospital Management System is essential for maintaining details about the Doctor and Patient. By using this system, a lot of time is saved.