DIGITAL HOSPITAL MANAGEMENT SYSTEM

NAME: METTU KARTHIK

FACE DETECTION USING WEBCAM

ABSTRACT

In this speedy world of medicine, it is a intimidating task to manage a multi talent hospital. A hospital management system is a computer that make easy to managing the work of the hospital or any medical set up.

This software will help in to building the entire work as paperless. It combines all the information as regards patients, doctors, staff, hospital administrative details and so on, into one software.

It has been sections for various professionals that make up a hospital

TABLE OF CONTENTS

LIST OF PLATES

1.INTRODUCTION

A good hospital management system would facilitate the expansion of the hospital network by easily managing resources, avoiding waste, creating an efficient workforce, and better data management. Such a system could provide hassle-free patient care and make everyone's job much easier. Hospital management software is a web-based system that works to manage different aspects of a medical facility. Its objective is to dematerialize the different processes of the medical center. It integrates all important data about doctors, patients, staff, administrative details, etc. There is a growing demand for healthcare app developers and for all the right reasons.

1. LITERATURE SURVEY

FHospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability.

HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals and HOSPITALs. More importantly it is backed by reliable and dependable support.

This HMS is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

A Hospital is a place where Patients come up for general diseases. Hospitals provide facilities like:-

- Consultation by Doctors on Diseases.
- Diagnosis for diseases.
- Providing treatment facility.
- Immunization for Patients/Children

Various operational works that are done in a Hospital are:-

- Recording information about the Patients that come.
- Generating bills.

• Recording information related to diagnosis given to Patients.

• Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff and Doctors. All these works are done on papers. The work is done as follows:-

• Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly. • Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up.

• Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.

• Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.

• Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

The limited time and resources have restricted us to incorporate, in this project, only main activities that are performed in a HOSPITAL Management System, but utmost care has been taken to make the system efficient and user friendly. "HOSPITAL Management System" has been designed to computerize the following functions that

are performed by the system:

1. Appointments for the Patients

23

a) Admission of New Patient

2. Free Medical Advice For the Patients

3. Discharge Detail Functions

b) Discharge of Patient

c) Doctor Assigning related to Patient's

Disease 4. Training Courses Provided by the

Hospital

5. Statement of Patient Details

1) Admitted Patient

2) Doctor Details

6. Total number of Patients admitted in the

Hospital 7. Doctors available in the Hospital

8. Preventive Health Checkups

9. Administrator Links

a.Login Form

b.To add new doctors in the site

c.List of patients

d.List of Doctors

3. SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

Operating System **:** Windows 10 and below,

Programming Language **:JAVA**

3.2 HARDWARE REQUIREMENTS

Hardware **:** Pentium Based System with a minimum of P4

RAM **:** 1GB(minimum)

Pentium 4

Pentium 4 is the intel processor that was released in the November 2000.The P4 processor has a viable clock speed that now exceeds 2GHz – as compared to the 1GHz of the Pentium3.

RAM

Random Access Memory is a form of computer storage that stores data and machine code currently being used. A RAM device allows data item to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory

4. PROJECT REQUIREMENTS

4.1 Functional Requirements

Functional requirements: *These requirements include processes like registration, report creation, database and check out.*

- **Registration requirements:** The registration process allows the hospital's front desk staff to add new patients to the system. It helps the hospital staff to provide a unique ID for each patient and then add them to the record sheet of the patient.
- **Report creation:** HMS helps generate reports on every patient, along with details that include phone number, bed number, patient's name, doctor assigned to the case, availability of bed etc.
- **Database:** It helps the hospital staff with all the mandatory information of the patient, their medical history, their full name, health number, postal code, etc.

4.2 Non Functional Requirements

Security: The HMS requires a patient identification system, help you with modification in the database, enable staff to add or view any patient's record, etc.

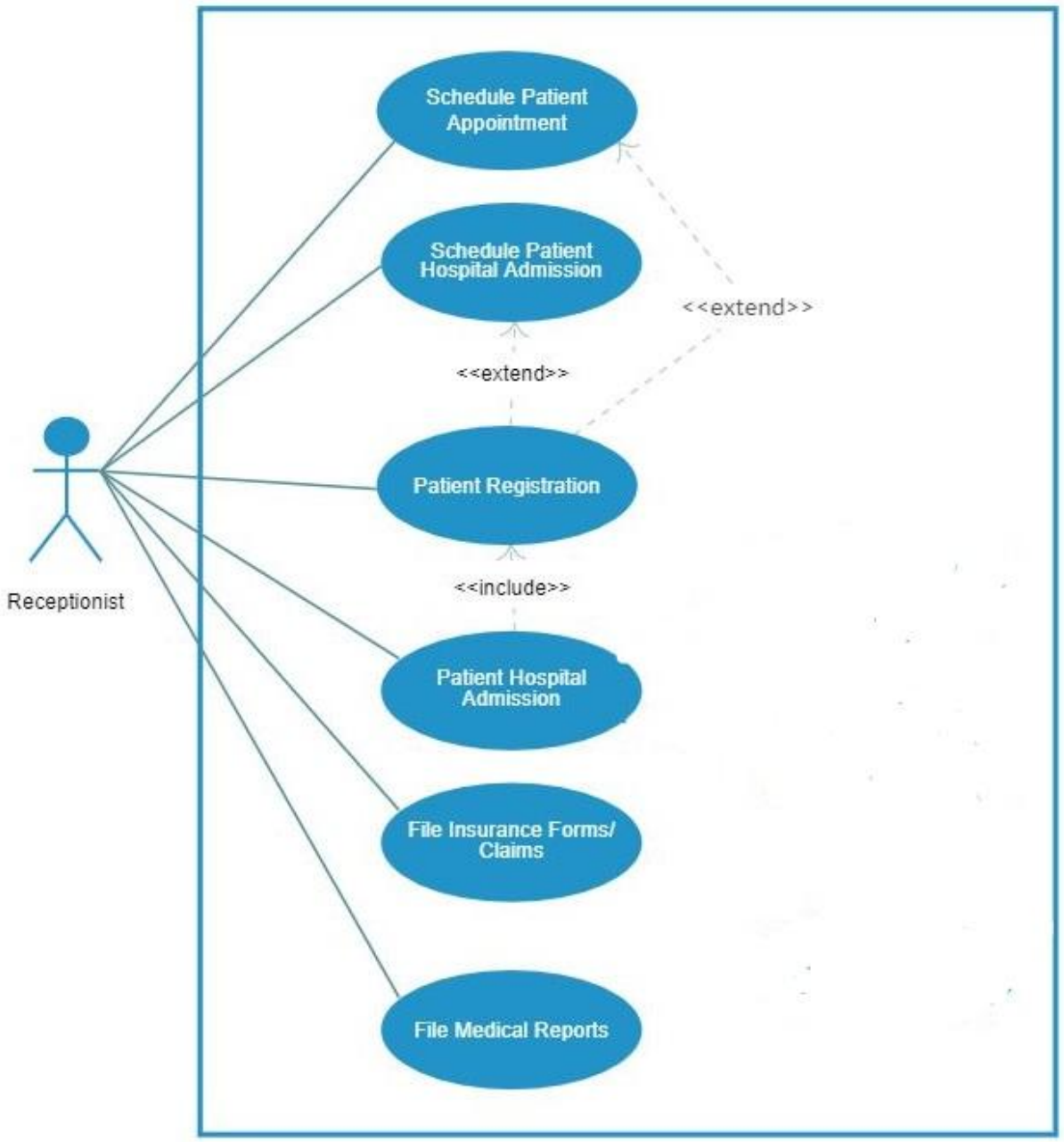
Reliability: Being a computer-operated system, it is available all the time.

Maintainability: The system offers efficiency for data backup, and check for errors in the patient's database.

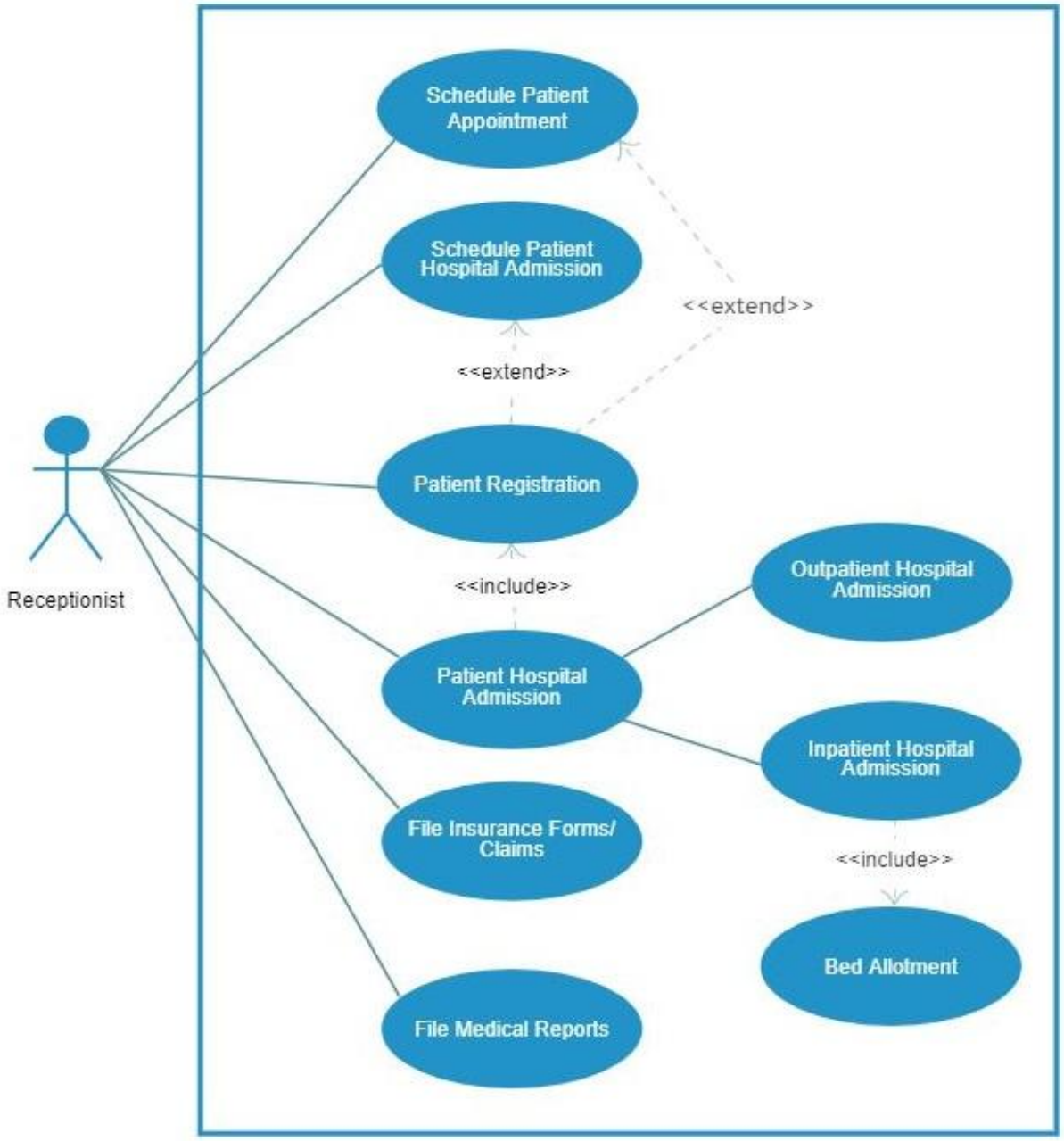
Performance: HMS helps you reduce the response time in with patients, enhances the capacity of handling patients by 1000 people, and ensures that guidelines are being followed

5. DESIGN

5.1 High Level Requirements



5.2 Low Level Requirements



6. IMPLEMENTATION

6.1 CODE SNIPPET

```
        import java.io.*;
import java.util.*;
import java.util.Calendar;
/* Hospital Management System Project in Java with Source Code PDF Visit for more
https://www.programmingwithbasics.com/ */
class staff
{
   String sid, sname, desg, sex;
   int salary;
   void new_staff()
   {
      Scanner input = new Scanner(System.in);
      System.out.print("id:-");
      sid = input.nextLine();
      System.out.print("name:-");
      sname = input.nextLine();
      System.out.print("desigination:-");
      desg = input.nextLine();
      System.out.print("sex:-");
      sex = input.nextLine();
      System.out.print("salary:-");
      salary = input.nextInt();
   }
   void staff_info()
   {
      System.out.println(sid + "\t" + sname + "\t" + sex + "\t" + salary);
   }
}
class doctor
{
   String did, dname, specilist, appoint, doc_qual;
   int droom;
   void new_doctor()
   {
      Scanner input = new Scanner(System.in);
      System.out.print("id:-");
      did = input.nextLine();
      System.out.print("name:-");
      dname = input.nextLine();
      System.out.print("specilization:-");
      specilist = input.nextLine();
      System.out.print("work time:-");
      appoint = input.nextLine();
      System.out.print("qualification:-");
```

```
        doc_qual = input.nextLine();
        System.out.print("room no.:-");
        droom = input.nextInt();
    }
    void doctor_info()
    {
        System.out.println(did + "\t" + dname + "  \t" + specilist + "     \t" + appoint + "    \t" + doc_qual + "
\t" + droom);
    }
}
class patient
{
    String pid, pname, disease, sex, admit_status;
    int age;
    void new_patient()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("id:-");
        pid = input.nextLine();
        System.out.print("name:-");
        pname = input.nextLine();
        System.out.print("disease:-");
        disease = input.nextLine();
        System.out.print("sex:-");
        sex = input.nextLine();
        System.out.print("admit_status:-");
        admit_status = input.nextLine();
        System.out.print("age:-");
        age = input.nextInt();
    }
    void patient_info()
    {
        System.out.println(pid + "\t" + pname + " \t" + disease + "     \t" + sex + "      \t" + admit_status + "\t"
+ age);
    }
}
class medical
{
    String med_name, med_comp, exp_date;
    int med_cost, count;
    void new_medi()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("name:-");
        med_name = input.nextLine();
        System.out.print("comp:-");
        med_comp = input.nextLine();
        System.out.print("exp_date:-");
        exp_date = input.nextLine();
        System.out.print("cost:-");
```

```
        med_cost = input.nextInt();
        System.out.print("no of unit:-");
        count = input.nextInt();
    }
    void find_medi()
    {
        System.out.println(med_name + "  \t" + med_comp + "    \t" + exp_date + "     \t" + med_cost);
    }
}
class lab
{
    String fecility;
    int lab_cost;
    void new_feci()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("fecility:-");
        fecility = input.nextLine();
        System.out.print("cost:-");
        lab_cost = input.nextInt();
    }
    void feci_list()
    {
        System.out.println(fecility + "\t\t" + lab_cost);
    }
}
class fecility //Sorry Facility but do not change the name
{
    String fec_name;
    void add_feci()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("fecility:-");
        fec_name = input.nextLine();
    }
    void show_feci()
    {
        System.out.println(fec_name);
    }
}
public class HMS
{
    public static void main(String args[])
    {
        String months[] = {
                "Jan",
                "Feb",
                "Mar",
                "Apr",
                "May",
```

```
            "Jun",
            "Jul",
            "Aug",
            "Sep",
            "Oct",
            "Nov",
            "Dec"
        };
        Calendar calendar = Calendar.getInstance();
        //System.out.println("--------------------------------------------------------------------------------");
        int count1 = 4, count2 = 4, count3 = 4, count4 = 4, count5 = 4, count6 = 4;
        System.out.println("\n--------------------------------------------------------------------------------");
        System.out.println("           *** Welcome to Hospital Management System Project in Java ***");
        System.out.println("--------------------------------------------------------------------------------");
        System.out.print("Date: " + months[calendar.get(Calendar.MONTH)] + " " +
calendar.get(Calendar.DATE) + " " + calendar.get(Calendar.YEAR));
        System.out.println("\t\t\t\t\t\tTime: " + calendar.get(Calendar.HOUR) + ":" +
calendar.get(Calendar.MINUTE) + ":" + calendar.get(Calendar.SECOND));
        doctor[] d = new doctor[25];
        patient[] p = new patient[100];
        lab[] l = new lab[20];
        fecility[] f = new fecility[20];
        medical[] m = new medical[100];
        staff[] s = new staff[100];
        int i;
        for (i = 0; i < 25; i++)
           d[i] = new doctor();
        for (i = 0; i < 100; i++)
           p[i] = new patient();
        for (i = 0; i < 20; i++)
           l[i] = new lab();
        for (i = 0; i < 20; i++)
           f[i] = new fecility();
        for (i = 0; i < 100; i++)
           m[i] = new medical();
        for (i = 0; i < 100; i++)
           s[i] = new staff();

        d[0].did = "21";
        d[0].dname = "Dr.Ghanendra";
        d[0].specilist = "ENT";
        d[0].appoint = "5-11AM";
        d[0].doc_qual = "MBBS,MD";
        d[0].droom = 17;
        d[1].did = "32";
        d[1].dname = "Dr.Vikram";
        d[1].specilist = "Physician";
        d[1].appoint = "10-3AM";
        d[1].doc_qual = "MBBS,MD";
        d[1].droom = 45;
```

```
d[2].did = "17";
d[2].dname = "Dr.Rekha";
d[2].specilist = "Surgeon";
d[2].appoint = "8-2AM";
d[2].doc_qual = "BDM";
d[2].droom = 8;
d[3].did = "33";
d[3].dname = "Dr.Pramod";
d[3].specilist = "Artho";
d[3].appoint = "10-4PM";
d[3].doc_qual = "MBBS,MS";
d[3].droom = 40;

p[0].pid = "12";
p[0].pname = "Pankaj";
p[0].disease = "Cancer";
p[0].sex = "Male";
p[0].admit_status = "y";
p[0].age = 30;
p[1].pid = "13";
p[1].pname = "Sumit";
p[1].disease = "Cold";
p[1].sex = "Male";
p[1].admit_status = "y";
p[1].age = 23;
p[2].pid = "14";
p[2].pname = "Alok";
p[2].disease = "Maleriya";
p[2].sex = "Male";
p[2].admit_status = "y";
p[2].age = 45;
p[3].pid = "15";
p[3].pname = "Ravi";
p[3].disease = "Diabetes";
p[3].sex = "Male";
p[3].admit_status = "y";
p[3].age = 25;

m[0].med_name = "Corex";
m[0].med_comp = "Cino pvt";
m[0].exp_date = "9-5-16";
m[0].med_cost = 55;
m[0].count = 8;
m[1].med_name = "Nytra";
m[1].med_comp = "Ace pvt";
m[1].exp_date = "4-4-15";
m[1].med_cost = 500;
m[1].count = 5;
m[2].med_name = "Brufa";
m[2].med_comp = "Reckitt";
```

```
m[2].exp_date = "12-7-17";
m[2].med_cost = 50;
m[2].count = 56;
m[3].med_name = "Pride";
m[3].med_comp = "DDF pvt";
m[3].exp_date = "12-4-12";
m[3].med_cost = 1100;
m[3].count = 100;

l[0].fecility = "X-ray      ";
l[0].lab_cost = 800;
l[1].fecility = "CT Scan    ";
l[1].lab_cost = 1200;
l[2].fecility = "OR Scan    ";
l[2].lab_cost = 500;
l[3].fecility = "Blood Bank";
l[3].lab_cost = 50;

f[0].fec_name = "Ambulance";
f[1].fec_name = "Admit Facility ";
f[2].fec_name = "Canteen";
f[3].fec_name = "Emergency";

s[0].sid = "22";
s[0].sname = "Prakash";
s[0].desg = "Technician";
s[0].sex = "Male";
s[0].salary = 5000;
s[1].sid = "23";
s[1].sname = "Komal";
s[1].desg = "Nurse";
s[1].sex = "Female";
s[1].salary = 2000;
s[2].sid = "24";
s[2].sname = "Raju";
s[2].desg = "Technician";
s[2].sex = "Male";
s[2].salary = 5000;
s[3].sid = "25";
s[3].sname = "Rani";
s[3].desg = "Nurse";
s[3].sex = "Female";
s[3].salary = 20000;

Scanner input = new Scanner(System.in);
int choice, j, c1, status = 1, s1 = 1, s2 = 1, s3 = 1, s4 = 1, s5 = 1, s6 = 1;
while (status == 1)
{
   System.out.println("\n                              MAIN MENU");
   System.out.println("--------------------------------------------------------------------------------");
```

```
System.out.println("1.Doctos 2. Patients 3.Medicines 4.Laboratories 5. Facilities 6. Staff ");
System.out.println("-----------------------------------------------------------------------------------");
choice = input.nextInt();
switch (choice)
{
   case 1:
   {
      System.out.println("-----------------------------------------------------------------------------------");
      System.out.println("                    **DOCTOR SECTION**");
      System.out.println("-----------------------------------------------------------------------------------");
      s1 = 1;
      while (s1 == 1)
      {
         System.out.println("1.Add New Entry\n2.Existing Doctors List");
         c1 = input.nextInt();
         switch (c1)
         {
            case 1:
            {
               d[count1].new_doctor();count1++;
               break;
            }
            case 2:
            {

System.out.println("-----------------------------------------------------------------------------------");
               System.out.println("id \t Name\t Specilist \t Timing \t Qualification \t Room No.");

System.out.println("-----------------------------------------------------------------------------------");
               for (j = 0; j < count1; j++)
               {
                  d[j].doctor_info();
               }
               break;
            }
         }
         System.out.println("\nReturn to Back Press 1 and for Main Menu Press 0");
         s1 = input.nextInt();
      }
      break;
   }
   case 2:
   {
      System.out.println("-----------------------------------------------------------------------------------");
      System.out.println("                    **PATIENT SECTION**");
      System.out.println("-----------------------------------------------------------------------------------");
      s2 = 1;
      while (s2 == 1)
      {
         System.out.println("1.Add New Entry\n2.Existing Patients List");
```

```
                c1 = input.nextInt();
                switch (c1)
                {
                   case 1:
                   {
                      p[count2].new_patient();count2++;
                      break;
                   }
                   case 2:
                   {

System.out.println("---------------------------------------------------------------------------------");
                      System.out.println("id \t Name \t Disease \t Gender \t Admit Status \t Age");

System.out.println("---------------------------------------------------------------------------------");
                      for (j = 0; j < count2; j++) {
                         p[j].patient_info();
                      }
                      break;
                   }
                }
                System.out.println("\nReturn to Back Press 1 and for Main Menu Press 0");
                s2 = input.nextInt();
             }
             break;
          }
          case 3:
          {
             s3 = 1;
             System.out.println("---------------------------------------------------------------------------------");
             System.out.println("                    **MEDICINE SECTION**");
             System.out.println("---------------------------------------------------------------------------------");
             while (s3 == 1)
             {
                System.out.println("1.Add New Entry\n2. Existing Medicines List");
                c1 = input.nextInt();
                switch (c1)
                {
                   case 1:
                   {
                      m[count3].new_medi();count3++;
                      break;
                   }
                   case 2:
                   {

System.out.println("---------------------------------------------------------------------------------");
                      System.out.println("Name \t Company \t Expiry Date \t Cost");

System.out.println("---------------------------------------------------------------------------------");
```

```
                    for (j = 0; j < count3; j++) {
                        m[j].find_medi();
                    }
                    break;
                }
            }
            System.out.println("\nReturn to Back Press 1 and for Main Menu Press 0");
            s3 = input.nextInt();
        }
        break;
    }
    case 4:
    {
        s4 = 1;
        System.out.println("-------------------------------------------------------------------------------");
        System.out.println("                    **LABORATORY SECTION**");
        System.out.println("-------------------------------------------------------------------------------");
        while (s4 == 1)
        {
            System.out.println("1.Add New Entry \n2.Existing Laboratories List");
            c1 = input.nextInt();
            switch (c1)
            {
                case 1:
                {
                    l[count4].new_feci();count4++;
                    break;
                }
                case 2:
                {

System.out.println("-------------------------------------------------------------------------------");
                    System.out.println("Fecilities\t\t Cost");

System.out.println("-------------------------------------------------------------------------------");
                    for (j = 0; j < count4; j++) {
                        l[j].feci_list();
                    }
                    break;
                }
            }
            System.out.println("\nReturn to Back Press 1 and for Main Menu Press 0");
            s4 = input.nextInt();
        }
        break;
    }
    case 5:
    {
        s5 = 1;
        System.out.println("-------------------------------------------------------------------------------");
```

```
            System.out.println("          **HOSPITAL FACILITY SECTION**");
            System.out.println("-------------------------------------------------------------------------------");
            while (s5 == 1)
            {
               System.out.println("1.Add New Facility\n2.Existing Fecilities List");
               c1 = input.nextInt();
               switch (c1)
               {
                  case 1:
                  {
                     f[count5].add_feci();count5++;
                     break;
                  }
                  case 2:
                  {

System.out.println("-------------------------------------------------------------------------------");
                     System.out.println("Hospital  Facility are:");

System.out.println("-------------------------------------------------------------------------------");
                     for (j = 0; j < count5; j++) {
                        f[j].show_feci();
                     }
                     break;
                  }
               }
               System.out.println("\nReturn to Back Press 1 and for Main Menu Press 0");
               s5 = input.nextInt();
            }
            break;
         }
         case 6:
         {
            s6 = 1;
            System.out.println("-------------------------------------------------------------------------------");
            System.out.println("                    **STAFF SECTION**");
            System.out.println("-------------------------------------------------------------------------------");
            while (s6 == 1)
            {
               String a = "nurse", b = "Technician", c = "security";
               System.out.println("1.Add New Entry \n2.Existing Nurses List\n3.Existing Technicians List
\n4.Existing Security List");
               c1 = input.nextInt();
               switch (c1)
               {
                  case 1:
                  {
                     s[count6].new_staff();count6++;
                     break;
                  }
```

```
                    case 2:
                    {

System.out.println("---------------------------------------------------------------------------");
                        System.out.println("id \t Name \t Gender \t Salary");

System.out.println("---------------------------------------------------------------------------");
                        for (j = 0; j < count6; j++)
                        {
                            if (a.equals(s[j].desg))
                                s[j].staff_info();
                        }
                        break;
                    }
                    case 3:
                    {

System.out.println("---------------------------------------------------------------------------");
                        System.out.println("id \t Name \t Gender \t Salary");

System.out.println("---------------------------------------------------------------------------");
                        for (j = 0; j < count6; j++)
                        {
                            if (b.equals(s[j].desg))
                                s[j].staff_info();
                        }
                        break;
                    }
                    case 4:
                    {

System.out.println("---------------------------------------------------------------------------");
                        System.out.println("id \t Name \t Gender \t Salary");

System.out.println("---------------------------------------------------------------------------");
                        for (j = 0; j < count6; j++)
                        {
                            if (c.equals(s[j].desg))
                                s[j].staff_info();
                        }
                        break;
                    }
                }
                System.out.println("\nReturn to Back Press 1 and for Main Menu Press 0");
                s6 = input.nextInt();
            }
            break;
        }
        default:
```

```
            {
                System.out.println(" You Have Enter Wrong Choice!!!");
            }
        }
        System.out.println("\nReturn to MAIN MENU Press 1");
        status = input.nextInt();
    }
  }
}
```

7. TESTING

High Level Test

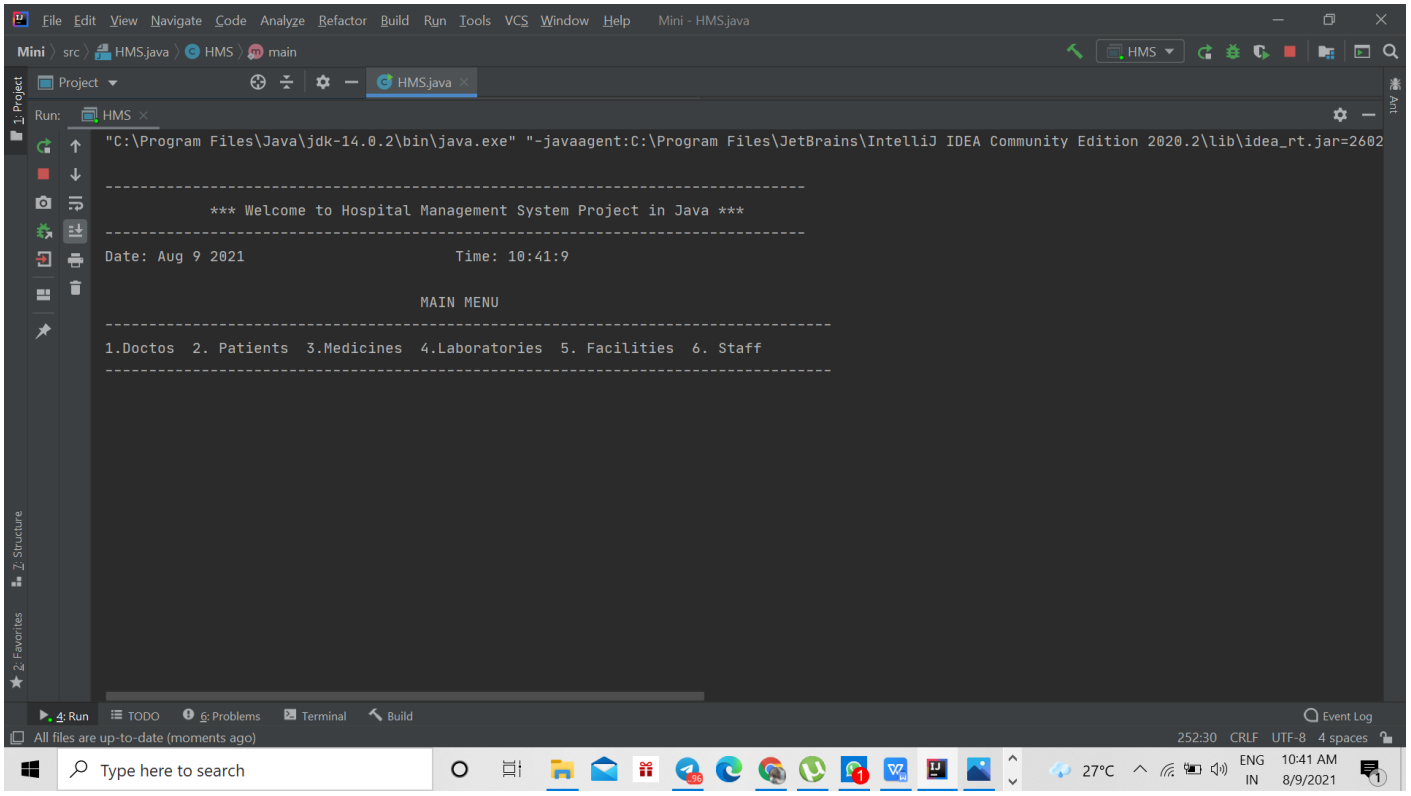
id	Description	Status
HR_01	Patient	Implemented
HR_02	Hospital	Implemented
HR_03	Staff	Implemented
HR_03	Advanced health equipment	Future
HR_04	New health problems	Future

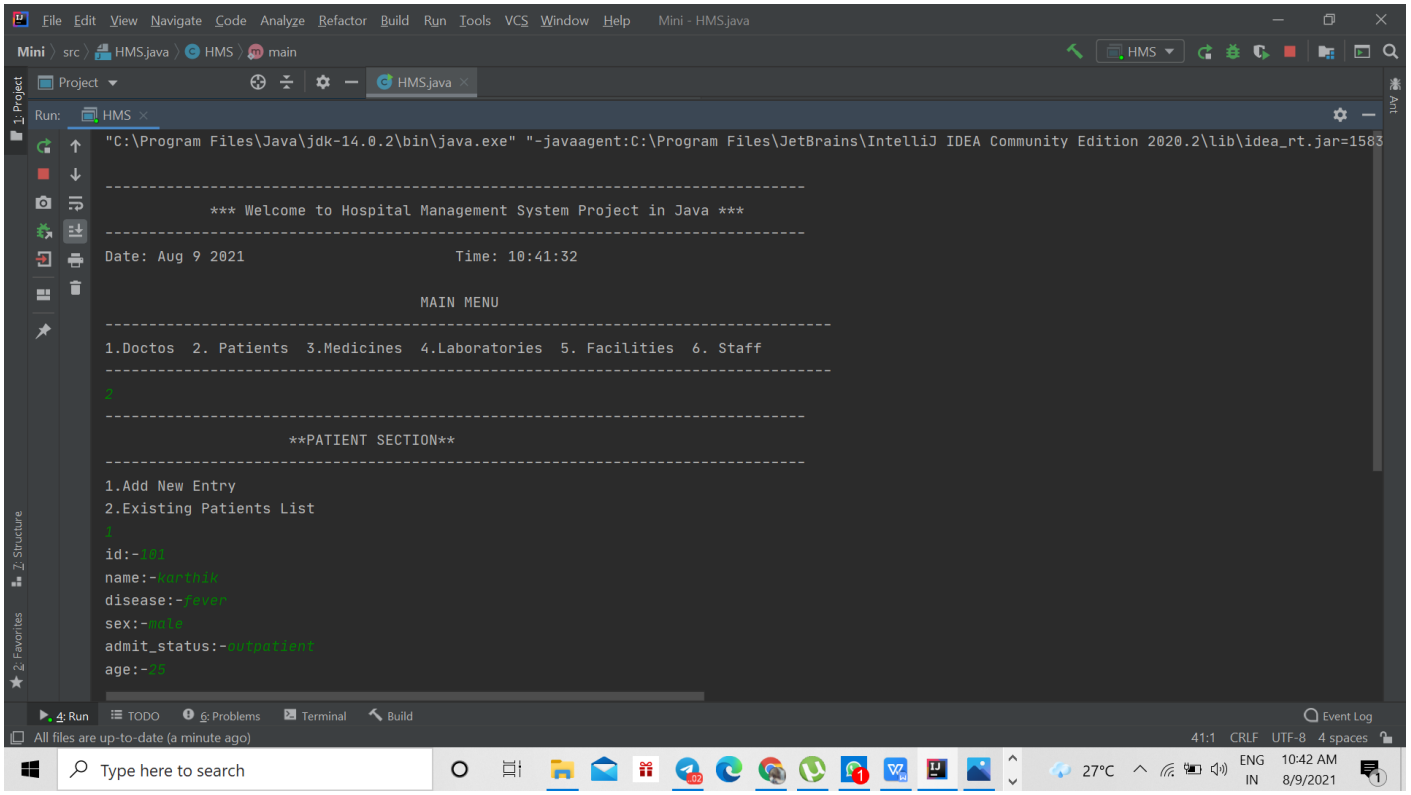
Low Level Test

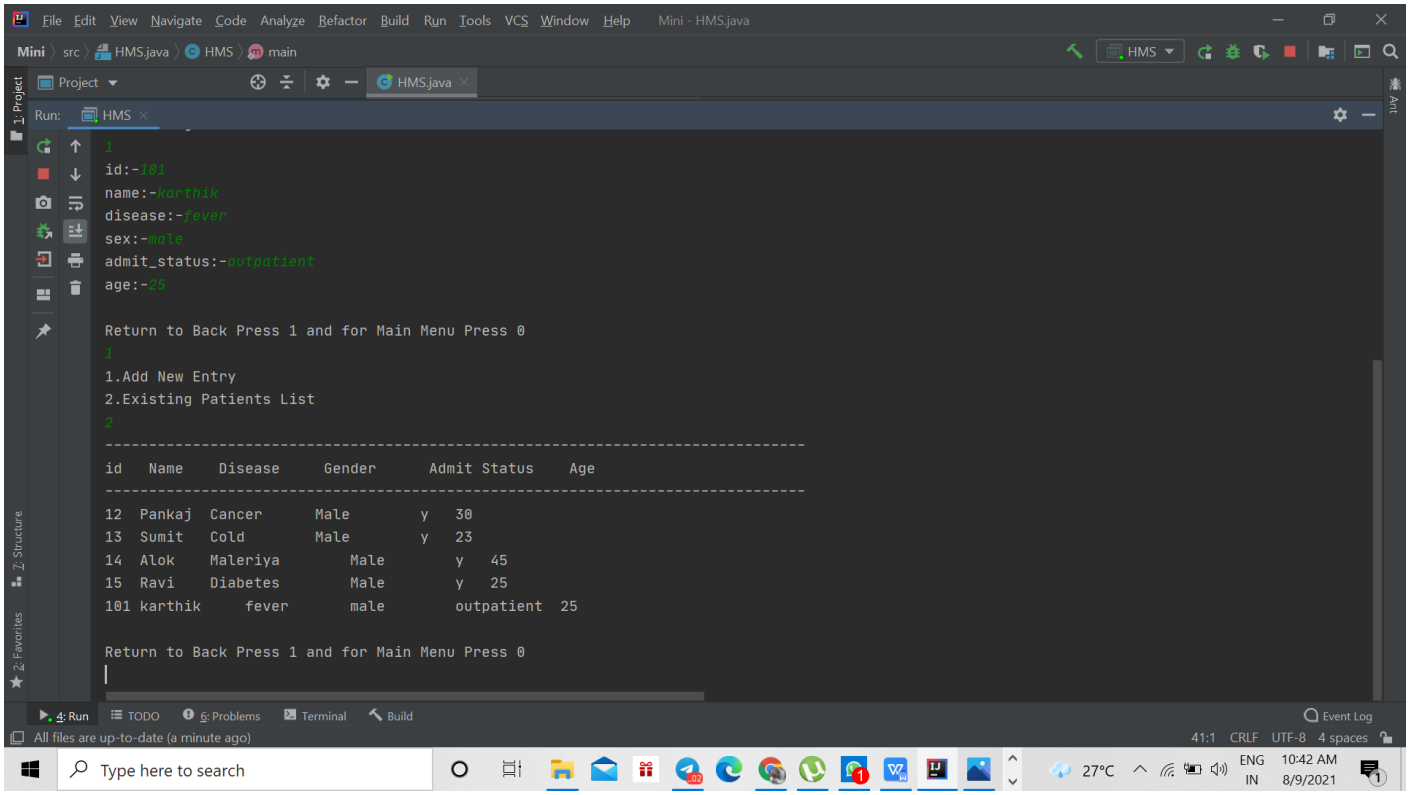
id	Description	Status
LR_01	Outpatient	Implemented
LR-02	Inpatient	Implemented
LR_03	Laboratories	Implemented
LR_04	Pharmacy	Implemented
LR_05	Operation Room	Future
LR_06	Doctor	Implemented
LR_07	Nurse	Implemented
LR_08	Technician	Implemented
LR_09	Security	Implemented
LR_10	Back up data	Future
LR_10	Update according to new health issues	Future

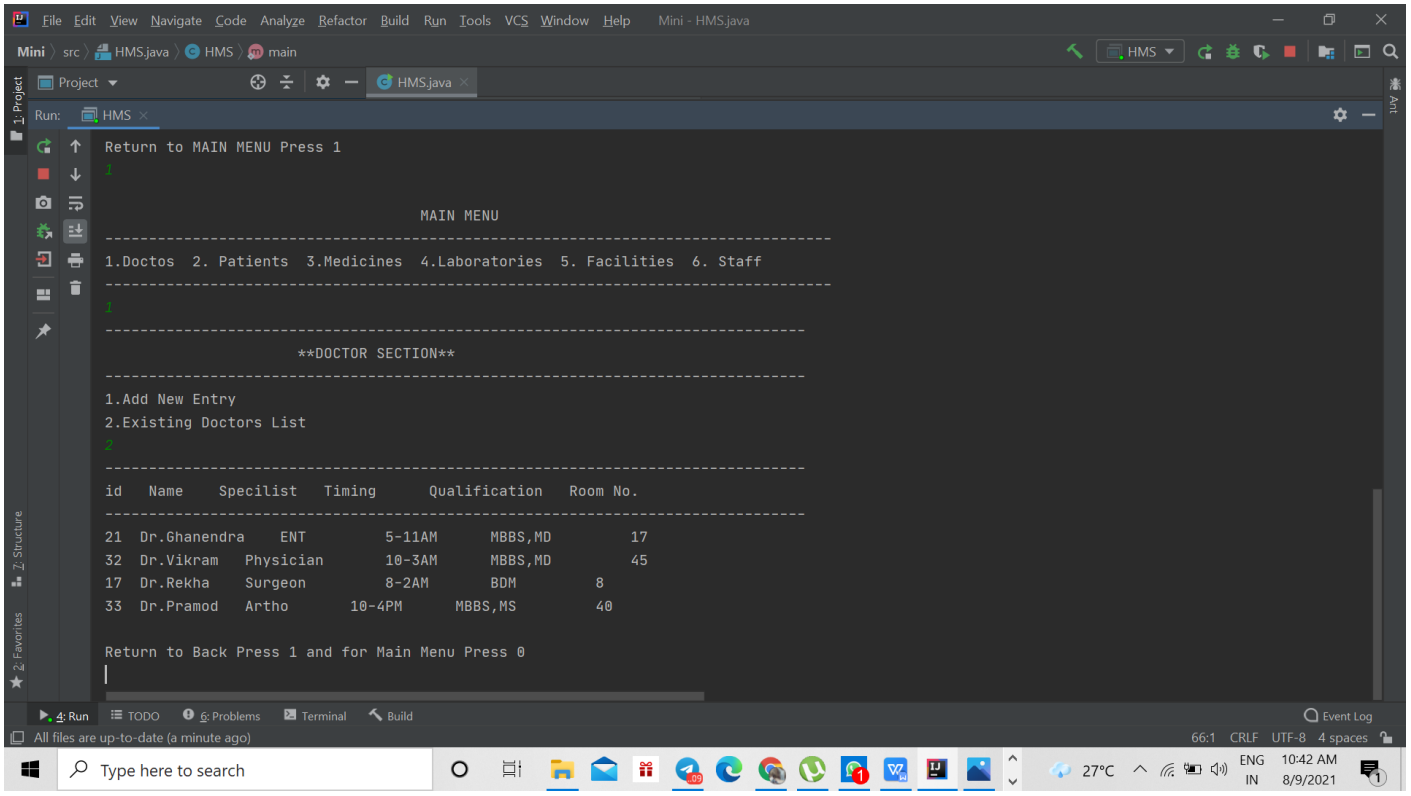
Note: These test cases are done manually by running the code.

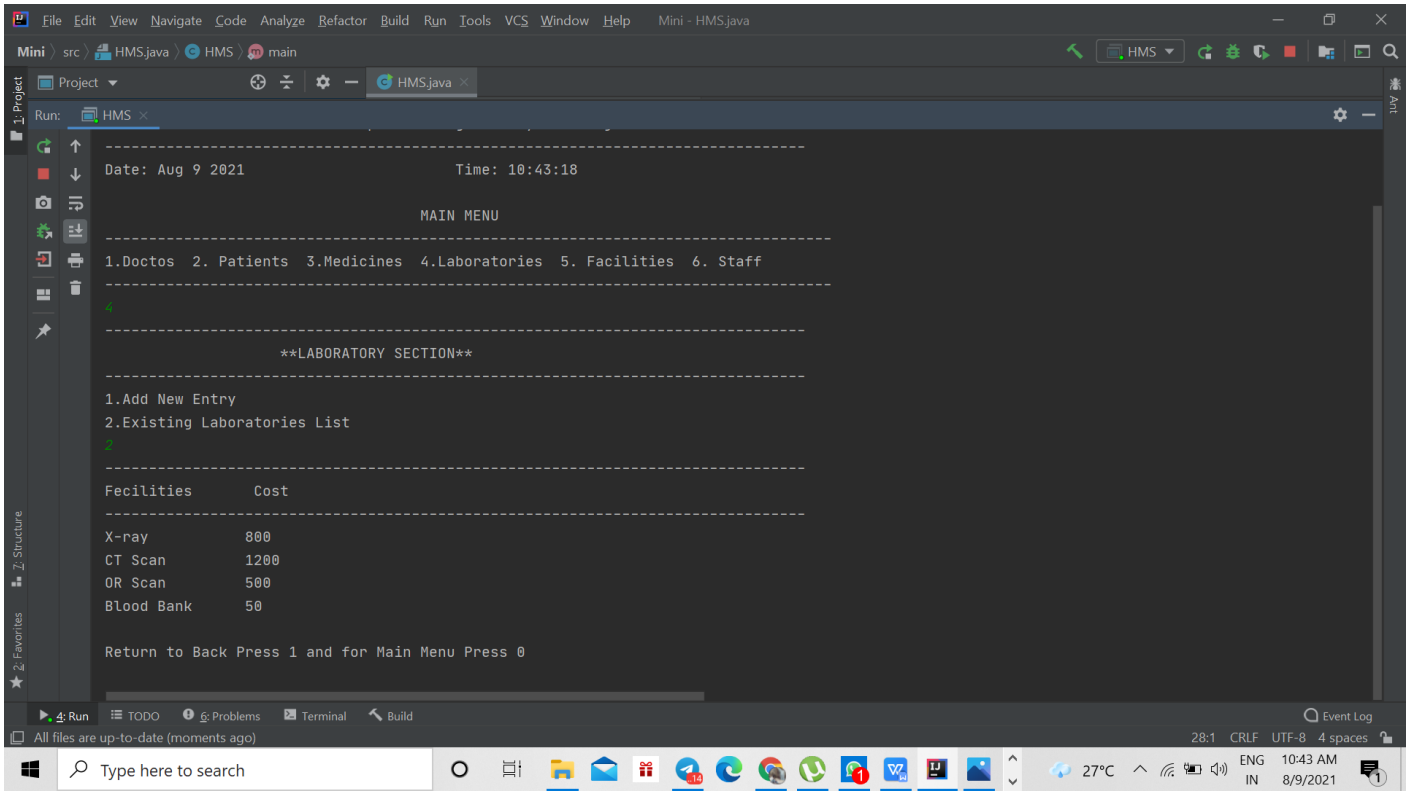
8. OUTPUT SCREENS






1
id:-101
name:-karthik
disease:-fever
sex:-male
admit_status:-outpatient
age:-25
Return to Back Press 1 and for Main Menu Press 0
1
1.Add New Entry
2.Existing Patients List
2
id Name Disease Gender Admit Status Age
12 Pankaj Cancer Male y 30
13 Sumit Cold Male y 23
14 Alok Maleriya Male y 45
15 Ravi Diabetes Male y 25
101 karthik fever male outpatient 25
Return to Back Press 1 and for Main Menu Press 0


Return to MAIN MENU Press 1
1
MAIN MENU
1.Doctos 2. Patients 3.Medicines 4.Laboratories 5. Facilities 6. Staff
1
DOCTOR SECTION
1.Add New Entry
2.Existing Doctors List
2
id Name Specilist Timing Qualification Room No.
21 Dr.Ghanendra ENT 5-11AM MBBS,MD 17
32 Dr.Vikram Physician 10-3AM MBBS,MD 45
17 Dr.Rekha Surgeon 8-2AM BDM 8
33 Dr.Pramod Artho 10-4PM MBBS,MS 40
Return to Back Press 1 and for Main Menu Press 0


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Mini - HMS.java
Mini src HMS.java HMS main
Project
HMS.java
Run: HMS
Date: Aug 9 2021 Time: 10:43:18
MAIN MENU
1.Doctos 2. Patients 3.Medicines 4.Laboratories 5. Facilities 6. Staff
4
LABORATORY SECTION
1.Add New Entry
2.Existing Laboratories List
2
Fecilities Cost
X-ray 800
CT Scan 1200
OR Scan 500
Blood Bank 50
Return to Back Press 1 and for Main Menu Press 0
4: Run
TODO
6: Problems
Terminal
Build
Event Log
All files are up-to-date (moments ago)
28:1 CRLF UTF-8 4 spaces
Type here to search
27°C
ENG IN
10:43 AM 8/9/2021

8.CONCLUSION

Hospital management system is all about the modernizing a hospital through use of technology. Computers helps in it and take over the manual system for quick and easy functioning. This hospital management system is a quite the reliable and is proven on many stages. All the basic requirements of the hospital are provided in the hospital in order to manage it perfectly and large amount of data can also be stored . It gives many facilities like searching for the detail of patient , billing facilities as well as the creation of test reports. So it;s a important system for modern days.

9. BIBLIOGRAPHY

1. https://medium.com/@physiokawal/hospital-management-system-hms-hospital-information-system-his-90adcec6fb36
2. http://www.peter-lo.com/Teaching/U08096/200912-B1