**PROJECT BASED LAB REPORT**

**On**

**HOSPITAL MANAGEMENT SYSTEM**

**Submitted in partial fulfilment of the**

**Requirements for the award of the Degree of**

**Bachelor of Technology**

**in**

**COMPUTER SCIENCE ENGINEERING**

**By**

**ROHINI PANDIRI 2100031934**



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**K L University**

Green Fields, Vaddeswaram, Guntur District-522 502

**2016-2017**

**K L University**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**



***CERTIFICATE***

This is to certify that this project based lab report entitled **“HOSPITAL MANAGEMENT SYSTEM”** is a bonafide work done by  **ANS NITHESH(150040025), Y.SURESH NAIDU(150040993)** in partial fulfilment of the requirements for the award of degree in **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE ENGINEERING**  during the Academic year 2016-2017.

**Faculty in Charge Head of the Department**

Smt.P.S.G Aruna Sri Dr.A.S.C.S.SASTRY

ASSOC.PROF PROFESSOR

**Project guide**

**BHUPESH DEKA**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**



***DECLARATION***

We hereby declare that this project based lab report titled **“HOSPITAL MANAGEMENT SYSTEM”** has been prepared by us in partial fulfilment of the requirements for the award of degree “**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGNEERING**” during the Academic year 2016-2017.

We also declare that this project based lab report is of our own efforts and it has not been submitted to any other university for the award of any degree.

ROHINI PANDIRI

2100031934

**ACKNOWLEDGEMENT**

Our sincere thanks to **Faculty name** in the Lab for their outstanding support through out the project for the successful completion of the work.

We express our gratitude to **DR.V. SRIKANTH,** Head of the Department for Computer science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project based Lab.

We would like to place on record the deep sense of gratitude to the honourable Vice Chancellor, K L University for providing the necessary facilities to carry the project based Lab.

Last, but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our project based Lab.

**ROHINI PANDIRI**

**2100031934**

**CONTENTS:**

**1.INTRODUCTION**

**2.MODUES DESCRIPTION**

**3.SOURECE CODE**

**4.RESULTS**

**5.CONCLUSION AND FUTURE SCOPE**

**ABSTRACT**

This project is the implementation for “ **HOSPITAL MANAGEMENT SYSTEM”**

is a program developed for managing details regarding number of patients

entering to the hospital. The program is very helpful to find the results of the inpatients

and outpatients easily. can search the doctors related to patients in hospital demo.

The project and implementation is developed by using java.

Java is a powerful tool for network programming and is the front-end of this

project. It is the most powerful database server and it provides data integrity and consistency.

**MODULES DESCRIPTION-**

**CHECKING THE NUMBER OF PATIENTS IN THE HOSPITAL:**

This helps us to provide the information regarding the number of

patients acquired by each member in the hospital

**FIND THE PATIENT WITH AVAILABILITY OF DOCTORS :**

This helps us to find out the information regarding the patient with

availability of doctors [if doctor gets more than 2n of the patients or

more than the remaining patients]

**CHECK OUTS MODULE:**

This helps us to print the information of the patient

with availability of doctors and check out with the permission of doctors

**INTRODUCTION**

Java is the object oriented programming language. It is interpreted and high level language. In Java, program is written by creating class and object. Unlike C, in which preprocessors and pointers are used, Java does not make use of these components. Any programming language that uses object oriented concepts like Encapsulation, Polymorphism, Inheritance, Abstraction is called object oriented programming language and JAVA comes under the same category.

**Features of Java program**

C is the widely used language. It provides a lot of features that are given below.

1. **Simple**

Java language is simple because its syntax is similar to C++. Similarly, it has removed many confusing and rarely used features like explicit pointers, operator overloading, etc. it is not needed to remove unreferenced objects because there is Automatic Garbage Collection in Java.

1. **Object Oriented**

Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior. The basic concepts of OOPs are: Object, Class, Inheritance, Polymorphism, Abstraction and Encapsulation.

1. **Platform independent**

Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

1. **Secured**

Java is secured because there is no use of explicit pointer and java programs run inside virtual machine sandbox.

1. **Robust**

Robust simply means strong. Java uses strong memory management. There is lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points make java robust.

1. **Architecture Neutral**

There are no implementation dependent features e.g. size of primitive types is fixed. Like- In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.

1. **Portable**

We may carry the java bytecode to any platform.

1. **High-performance**

Java is faster than traditional interpretation but still somewhat slower than a compiled language like C.

1. **Distributed**

Wecan create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

**10) Multi-threaded**

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area.

**PROJECT DESCRIPTION**

The project is done based on java programming. This is through object oriented programming. This gives some necessary information about Hospitals. When we want to search any information regarding doctors, patients and treatment records, it is very difficult since we have to refer many registers for collecting the relevant

information it takes lots of time. we can get all the information of any doctor and patient immediately and accurately. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy .some of the issues can be solved easily by using this project.

The project Hospital Management system deals with the information of every doctor, patients and treatment records includes writing the record, updating the record, searching the record, displaying the records, and quitting of that particular module and so on....

**AIM**

* 1. To develop a system for providing Hospital Management system.
  2. To develop a java code for Hospital Management system.
  3. To make student understand the various concepts associated java such as Constructor, Method Overloading, Method Overriding, final keyword, this keyword, Access Specifies, Exception Handling such as throw and Throws, files etc..

**ADVANTAGES:**

* 1. This program is required to understand the various concepts associated java such as Constructor, Method Overloading, Method Overriding, final keyword, this keyword, Access Specifies, Exception Handling such as throw and Throws, files etc..
* One who prepares this code themselves can understand what are the basic errors that we can get while writing any program and how to debug them.

**DISADVANTAGES:**

* It takes time to write and execute the program.

**FUTURE ENHANCEMENTS:**

* This program can be extended not only to write, update and search records but also to add user id, view my records, logout from the system and so on.

**SYSTEM REQUIREMENTS:**

**Software Requirements**

The major software requirements of the project are as follows:

Language : Java SE

Operating system**:** Windows Xp or later.

**Hardware Requirements**

The hardware requirements that map towards the software are as follows:

RAM : 8GB

Processor : Intel

Mouse

Keyboard.

**JAVA CODE:**

It is very difficult thing to search the record of doctors, patients, and treatments done in Hospital. and also the main Concept is to storing in a java program and not to refer any registers etc..This code is having the username and password to enter the menu....

import java.io.\*;

import java.util.Scanner;

class Patientrec {

String pname,pgender,pbloodgroup;

int page,pphonenumber,spid,pid;

public int q[]=new int[100];

String str1="";

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

public void displayprecord() {

System.out.println("\t #####Patient Details#####");

System.out.print("I.D. Number:"+pid);

System.out.print("\tName:"+pname);

System.out.print("\tAge:"+page);

System.out.print("\tGender:"+pgender);

System.out.print("\tBlood Group:"+pbloodgroup);

System.out.print("\tPhone Number:"+pphonenumber);

System.out.println("\n");

}

public void pim() throws IOException {

int a,k,x,i=0;

boolean flag1=true;

while(flag1)

{

System.out.println("1. Write Record ");

System.out.println("2. Update Record ");

System.out.println("3. Search");

System.out.println("4. Display Record");

System.out.println("5. Quit");

System.out.print("Enter your choice: ");

a= Integer.parseInt(br.readLine());

switch(a)

{

case 1: // Creation of Patient file

FileWriter fout1 = new FileWriter("patient.txt");

//Read data from Keyboard

System.out.println("How many Patients?");

x= Integer.parseInt(br.readLine());

for(k=0;k<x;k++)

{

System.out.print("Enter Patient Name:");

pname=br.readLine();

System.out.print("Enter Phone Number:");

pphonenumber= Integer.parseInt(br.readLine());

System.out.print("Enter Age:");

page =Integer.parseInt(br.readLine());

System.out.print("Enter Gender:");

pgender=br.readLine();

System.out.print("Enter Bloodgroup:");

pbloodgroup=br.readLine();

System.out.print("Enter I.D. Number:");

pid=Integer.parseInt(br.readLine());

q[i] = pid;i++;

// Written to Patient file

fout1.write(pphonenumber+" \n");

fout1.write(pname+" \n");

fout1.write(page+" \n");

fout1.write(pgender+" \n");

fout1.write(pbloodgroup+" \n");

fout1.write(pid+" \n\n");

}

fout1.write("EOF");

fout1.close();

System.out.println("\nFile is created\n");

break;

case 2: // Updation of Patient file

FileWriter fout2 = new FileWriter("patient.txt",true);

//Read data from Keyboard

System.out.println("How many Patients?");

x= Integer.parseInt(br.readLine());

for(k=0;k<x;k++)

{

System.out.print("Enter Patient Name:");

pname=br.readLine();

System.out.print("Enter Phone Number:");

pphonenumber= Integer.parseInt(br.readLine());

System.out.print("Enter Age:");

page =Integer.parseInt(br.readLine());

System.out.print("Enter Gender:");

pgender=br.readLine();

System.out.print("Enter Bloodgroup:");

pbloodgroup=br.readLine();

System.out.print("Enter I.D. Number:");

pid=Integer.parseInt(br.readLine());

q[i] = pid;i++;

// Written to Patient file

fout2.write(pphonenumber+" \n");

fout2.write(pname+" \n");

fout2.write(page+" \n");

fout2.write(pgender+" \n");

fout2.write(pbloodgroup+" \n");

fout2.write(pid+" \n\n");

}

fout2.write("EOF");

fout2.close();

System.out.println("\nFile is updated\n");

break;

case 3: System.out.print("Enter Patient I.D. Number:");

spid = Integer.parseInt(br.readLine());

case 4: FileReader fin = new FileReader("patient.txt");

Scanner sc = new Scanner(fin);

while(sc.hasNextInt())

{

//Read data from file

pphonenumber = sc.nextInt();

pname = sc.next();

page = sc.nextInt();

pgender = sc.next();

pbloodgroup = sc.next();

pid = sc.nextInt();

q[i] = pid;

if(a==4)

displayprecord();

else if(a==3){

if(spid == q[i])

displayprecord();

else

System.out.println("Record not found");

}

}

str1 = sc.next();

if (str1.equals("EOF")) {

System.out.println("End of file");

} else {

System.out.println("File format error.");

}

fin.close();

break;

case 5: flag1=false;

break;

default:System.out.println("Wrong Choice!!");

flag1=false;

}

}

}

}

class Doctorrec {

String dname,dgender,dspecial;

int dage,dphonenumber,sdid,did;

public int r[]=new int[100];

String str2="";

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

public void displaydrecord() {

System.out.println("\t #####Doctor Details#####");

System.out.print("I.D. Number:"+did);

System.out.print("\tName:"+dname);

System.out.print("\tAge:"+dage);

System.out.print("\tGender:"+dgender);

System.out.print("\tBlood Group:"+dspecial);

System.out.print("\tPhone Number:"+dphonenumber);

System.out.println("\n");

}

public void dim() throws IOException {

int b,l,y;

boolean flag2=true;

while(flag2)

{

System.out.println("1. Write Record ");

System.out.println("2. Update Record ");

System.out.println("3. Search");

System.out.println("4. Display Record");

System.out.println("5. Quit");

System.out.print("Enter your choice: ");

b= Integer.parseInt(br.readLine());

switch(b)

{

case 1: // Creation of Doctor file

FileWriter fout1 = new FileWriter("doctor.txt");

System.out.println("How many Doctors?\t");

y= Integer.parseInt(br.readLine());

for(l=0;l<y;l++)

{

System.out.print("Enter Doctor Name:");

dname=br.readLine();

System.out.print("Enter Phone Number:");

dphonenumber= Integer.parseInt(br.readLine());

System.out.print("Enter Age:");

dage =Integer.parseInt(br.readLine());

System.out.print("Enter Gender:");

dgender=br.readLine();

System.out.print("Enter Specialisation:");

dspecial=br.readLine();

System.out.print("Enter I.D. Number:");

did=Integer.parseInt(br.readLine());

// Written to Doctor file

fout1.write(dphonenumber+" \n");

fout1.write(dname+" \n");

fout1.write(dage+" \n");

fout1.write(dgender+" \n");

fout1.write(dspecial+" \n");

fout1.write(did+" \n\n");

}

fout1.write("EOF");

fout1.close();

System.out.println("\nFile is created\n");

break;

case 2: // Updation of Doctor file

FileWriter fout2 = new FileWriter("doctor.txt",true);

System.out.println("How many Doctors?\t");

y= Integer.parseInt(br.readLine());

for(l=0;l<y;l++)

{

System.out.print("Enter Doctor Name:");

dname=br.readLine();

System.out.print("Enter Phone Number:");

dphonenumber= Integer.parseInt(br.readLine());

System.out.print("Enter Age:");

dage =Integer.parseInt(br.readLine());

System.out.print("Enter Gender:");

dgender=br.readLine();

System.out.print("Enter Specialisation:");

dspecial=br.readLine();

System.out.print("Enter I.D. Number:");

did=Integer.parseInt(br.readLine());

// Written to Doctor file

fout2.write(dphonenumber+" \n");

fout2.write(dname+" \n");

fout2.write(dage+" \n");

fout2.write(dgender+" \n");

fout2.write(dspecial+" \n");

fout2.write(did+" \n\n");

}

fout2.write("EOF");

fout2.close();

System.out.println("\nFile is updated\n");

break;

case 3: System.out.print("Enter Doctor I.D. Number:");

sdid = Integer.parseInt(br.readLine());

case 4: FileReader fin = new FileReader("doctor.txt");

Scanner sc = new Scanner(fin);

while(sc.hasNextInt())

{

//Read data from file

dphonenumber = sc.nextInt();

dname = sc.next();

dage = sc.nextInt();

dgender = sc.next();

dspecial = sc.next();

did = sc.nextInt();

int i=0;

r[i]=did;i++;

if(b==4)

displaydrecord();

else if(b==3){

if(sdid == did)

displaydrecord();

else

System.out.println("Record not found");

}

}

str2 = sc.next();

if (str2.equals("EOF")) {

System.out.println("End of file");

} else {

System.out.println("File format error.");

}

fin.close();

break;

case 5: flag2=false;

break;

default:System.out.println("Wrong Choice!!");

flag2=false;

break;

}

}

}

}

class Treatmentrec {

String dateofjoining,treatment,diagnoseddisease,prescription,dischargedate,result;

int tid,stid,spid,sdid,t1,t2;

boolean f=true;

String str3="";

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

Patientrec p1 = new Patientrec();

Doctorrec d1 = new Doctorrec();

void displaytrecord() throws IOException {

System.out.print("Treament I.D. Number:"+tid);

System.out.print("Date of Joining:"+dateofjoining);

System.out.print("Patient's I.D. Number:"+spid);

System.out.print("Doctor's I.D. Number:"+sdid);

System.out.print("Treament:"+treatment);

System.out.print("Disease Diagnosed with:"+diagnoseddisease);

System.out.print("Medicines Prescription:"+prescription);

System.out.print("Date of Discharge:"+dischargedate);

System.out.print("RESULT:"+result);

System.out.println("\n");

}

public void tim() throws IOException {

int c,m,z;

boolean flag3=true;

while(flag3)

{

System.out.println("1. Write Record ");

System.out.println("2. Update Record ");

System.out.println("3. Search");

System.out.println("4. Display Record");

System.out.println("5. Quit");

System.out.print("Enter your choice: ");

c= Integer.parseInt(br.readLine());

switch(c)

{

case 1: FileWriter fout3 = new FileWriter("treatment.txt");

System.out.print("How many Treatment Records? ");

z= Integer.parseInt(br.readLine());

for(m=0;m<z;m++)

{

System.out.print("Enter Date of Joining:");

dateofjoining=br.readLine();

System.out.print("Enter Patient's I.D. Number:");

t1= Integer.parseInt(br.readLine());

int k=0;

if(p1.q.length!=0) {

if(t1 == p1.q[k])

System.out.println("Patient I.D number not found");

else

{

t1=1;

spid=t1;k++;

System.out.println("Patient I.D number found");

}

}

System.out.print("Enter Doctor's I.D. Number:");

t2=Integer.parseInt(br.readLine());

int l=0;

if(d1.r.length!=0){

if(d1.r[l] == t2)

System.out.println("Doctor I.D number not found");

else{

t2=2;l++;

sdid= t2;

System.out.println("Doctor I.D number found");

}

}

System.out.print("Enter Treament:");

treatment=br.readLine();

System.out.print("Enter Disease Diagnosed with:");

diagnoseddisease=br.readLine();

System.out.print("Enter Medicines Prescribed:");

prescription=br.readLine();

System.out.print("Enter Date of Discharge:");

dischargedate=br.readLine();

System.out.print("Enter RESULT:");

result=br.readLine();

System.out.print("Enter Treatment I.D. Number:");

tid =Integer.parseInt(br.readLine());

//Write to file.

fout3.write(tid+" \n");

fout3.write(dateofjoining+" \n");

fout3.write(treatment+" \n");

fout3.write(diagnoseddisease+" \n");

fout3.write(prescription+" \n");

fout3.write(dischargedate+" \n");

fout3.write(result+" \n");

fout3.write(sdid+" \n\n");

}

fout3.write("EOF");

fout3.close();

System.out.println("File is created");

break;

case 2: FileWriter fout4 = new FileWriter("treatment.txt");

System.out.print("How many Treatment Records? ");

z= Integer.parseInt(br.readLine());

for(m=0;m<z;m++)

{

System.out.print("Enter Date of Joining:");

dateofjoining=br.readLine();

System.out.print("Enter Patient's I.D. Number:");

t1= Integer.parseInt(br.readLine());

int k=0;

if(p1.q.length!=0) {

if(t1 == p1.q[k])

System.out.println("Patient I.D number not found");

else

{

t1=1;

spid=t1;k++;

System.out.println("Patient I.D number found");

}

}

System.out.print("Enter Doctor's I.D. Number:");

t2=Integer.parseInt(br.readLine());

int l=0;

if(d1.r.length!=0){

if(d1.r[l] == t2)

System.out.println("Doctor I.D number not found");

else{

t2=2;l++;

sdid= t2;

System.out.println("Doctor I.D number found");

}

}

System.out.print("Enter Treament:");

treatment=br.readLine();

System.out.print("Enter Disease Diagnosed with:");

diagnoseddisease=br.readLine();

System.out.print("Enter Medicines Prescribed:");

prescription=br.readLine();

System.out.print("Enter Date of Discharge:");

dischargedate=br.readLine();

System.out.print("Enter RESULT:");

result=br.readLine();

System.out.print("Enter Treatment I.D. Number:");

tid =Integer.parseInt(br.readLine());

//Write to file.

fout4.write(tid+" \n");

fout4.write(dateofjoining+" \n");

fout4.write(treatment+" \n");

fout4.write(diagnoseddisease+" \n");

fout4.write(prescription+" \n");

fout4.write(dischargedate+" \n");

fout4.write(result+" \n");

fout4.write(sdid+" \n\n");

}

fout4.write("EOF");

fout4.close();

System.out.println("File is created");

break;

case 3:System.out.print("Enter Treatment I.D. Number : ");

stid = Integer.parseInt(br.readLine());

case 4:FileReader fin = new FileReader("treatment.txt");

Scanner sc = new Scanner(fin);

while(sc.hasNextInt())

{

//Read data from file

tid = sc.nextInt();

dateofjoining = sc.next();

treatment = sc.next();

diagnoseddisease = sc.next();

prescription = sc.next();

dischargedate = sc.next();

result = sc.next();

spid = sc.nextInt();

sdid = sc.nextInt();

if(c==3)

{

displaytrecord();

}

if(c==2)

{

if(spid==p1.pid && sdid==d1.did && stid==tid)

{

displaytrecord();

}

}

}

str3= sc.next();

if(str3.equals("EOF"))

{

System.out.println("\nEnd of file");

}

else

{

System.out.println("File format error.");

}

fin.close();

break;

case 5: flag3=false;

break;

default:System.out.println("Wrong Choice!!");

}

}

}

}

public class Project

{

public static void main(String[] args) throws IOException {

int choice;

boolean flag=true;

final int LIMIT = 300;

int count = 0;

String username1 = "hospital";

String password1 = "management";

String username2,password2;

Scanner scan = new Scanner (System.in);

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

Patientrec pr = new Patientrec();

Doctorrec dr = new Doctorrec();

Treatmentrec tr = new Treatmentrec();

while(flag){

System.out.print ("Enter Username: ");

username2 = scan.nextLine();

if (username1.equals(username2)){

while(count < LIMIT){

System.out.print("Enter Passwsord: ");

password2 = scan.nextLine();

if(password2.equals(password1))

{

System.out.println("Welcome!");

System.out.println("\tMenu");

System.out.println("1. Patient Information Management ");

System.out.println("2. Doctor Information Management");

System.out.println("3. Treatment Information Management");

System.out.println("4. Quit");

System.out.print("Enter your choice: ");

choice= Integer.parseInt(br.readLine());

switch(choice){

case 1:

pr.pim();

break;

case 2:

dr.dim();

break;

case 3:

tr.tim();

break;

case 4:

System.out.println("Quit");

System.exit(0);

default:System.out.println("Wrong Choice!!");

}

}

else{

System.out.println("Incorrect entry. Please try again.");

count++;

}

}

System.out.println("Your 3 tries are up!");

System.out.println("Wiping hard drive!");

System.exit(0);

}

}

}

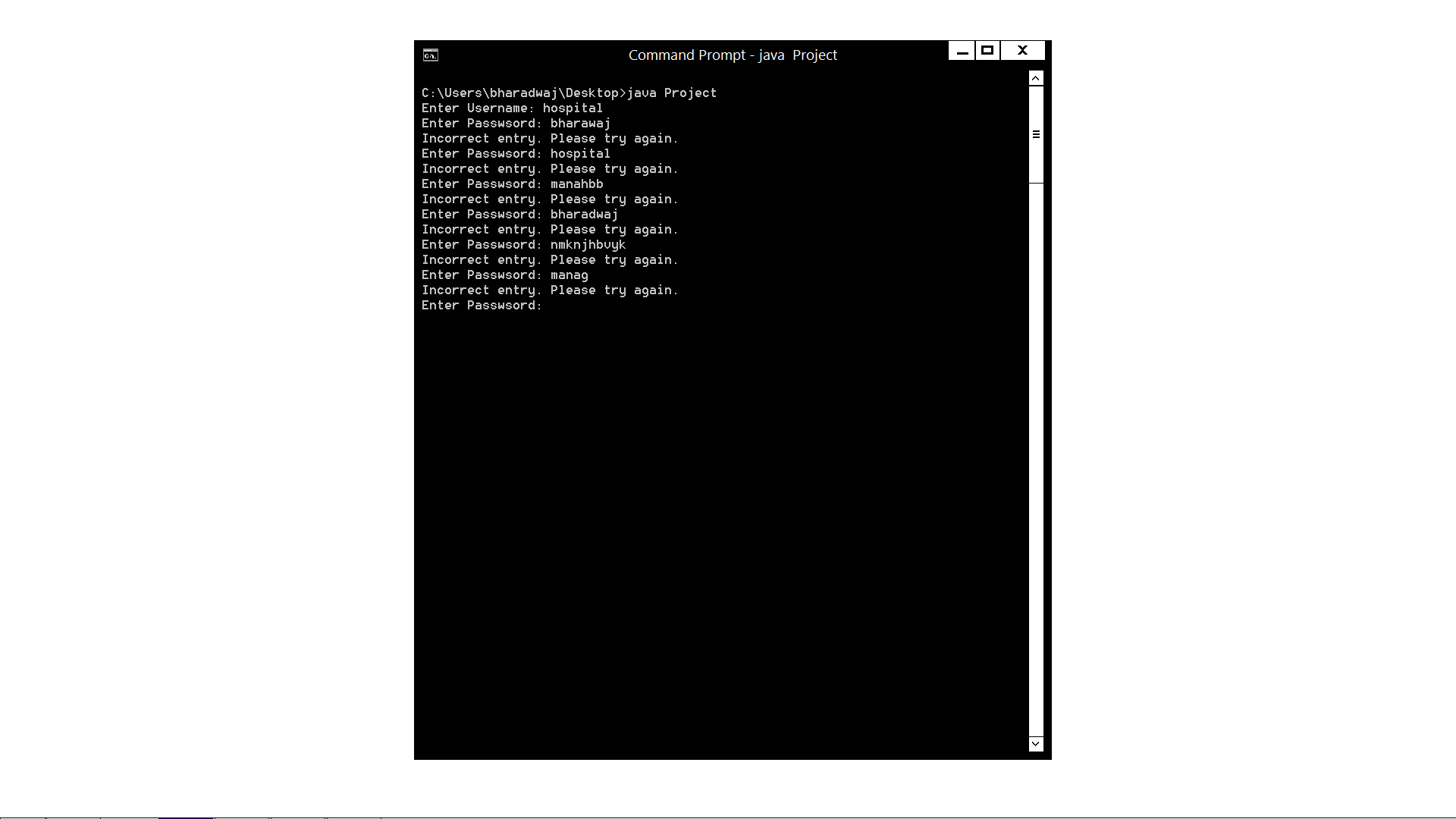
}

**SCREEN SHOTS:**

**Screen shot for username and password:**

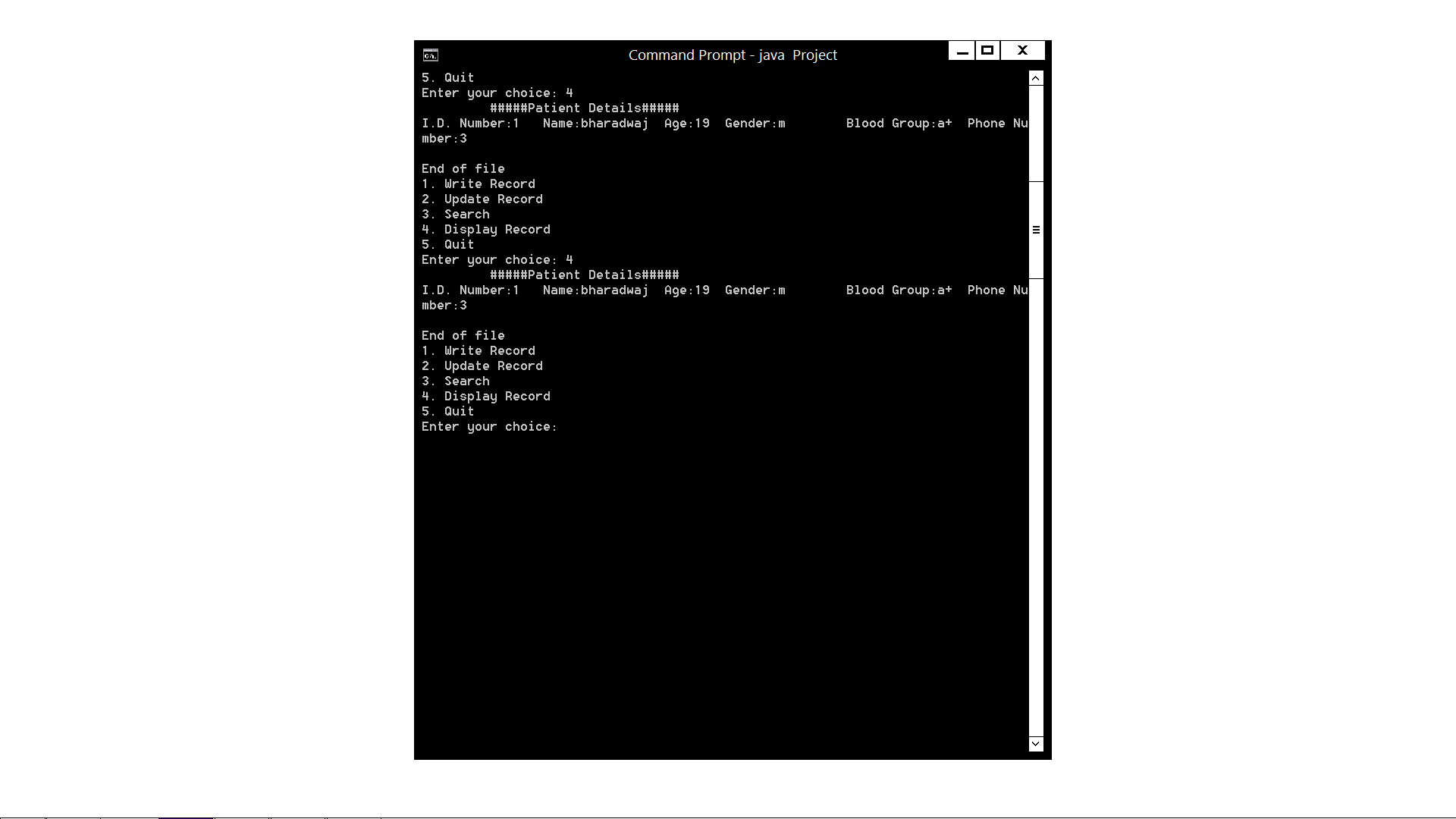
****

**if username and password are incorrect**

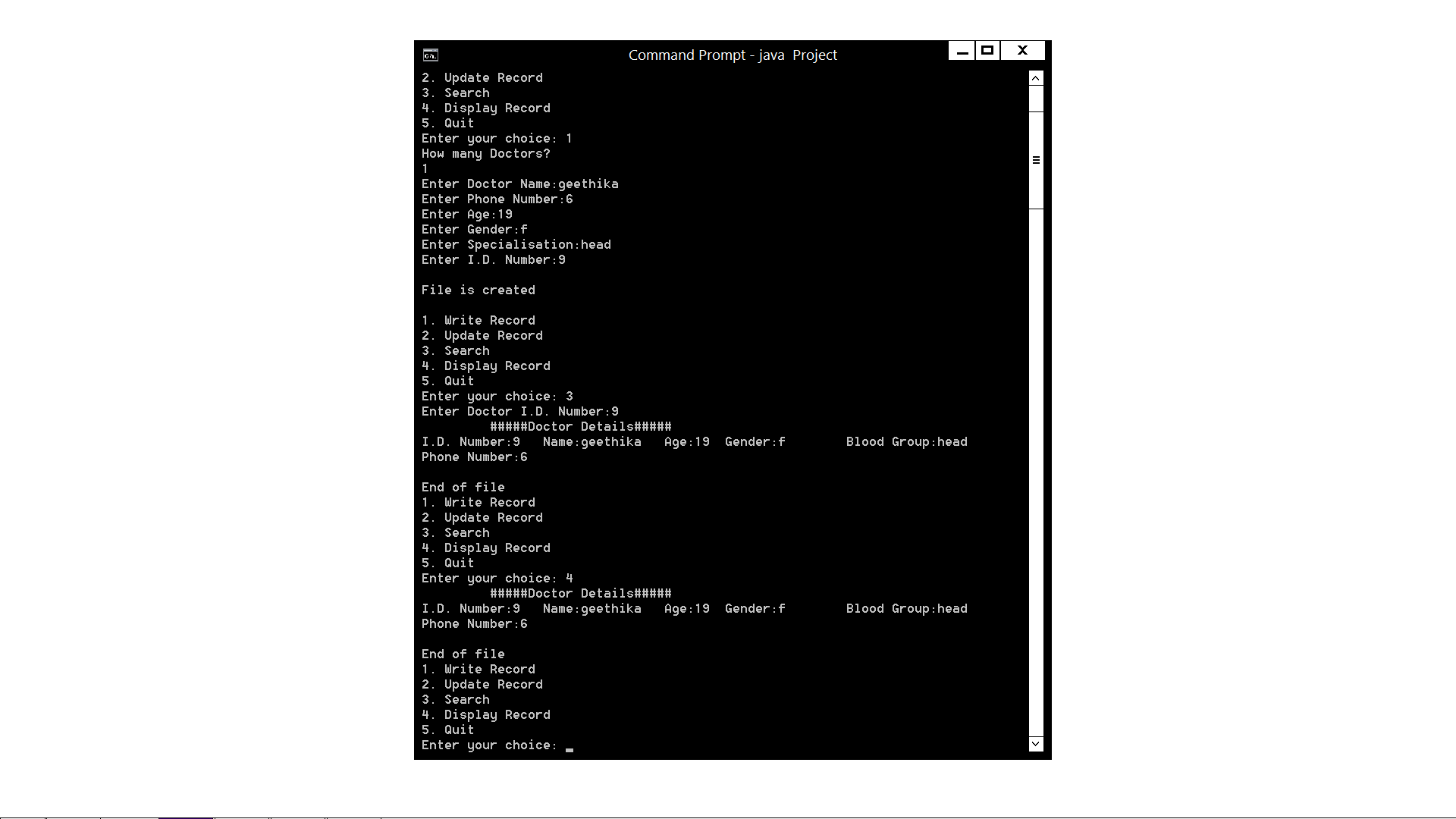
****

**screen shot for patient information system**

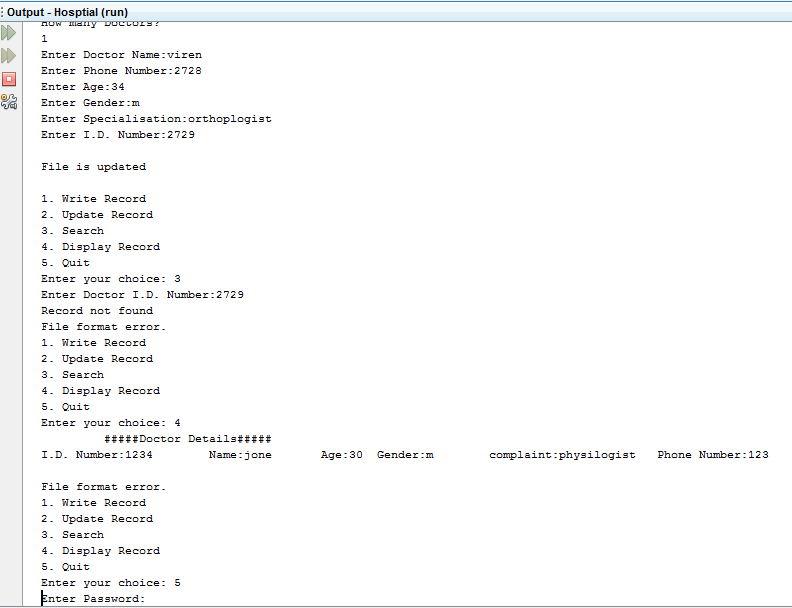
****

****

**screen shot for doctor information**

****

**screen shot for treatment information:**

****

**CONCLUSION:**

Overall, this system is user friendly and accurate that is available for getting hospital information at the time of requirement. It simplifies the task and reduces paperwork. This system is more accurate, user friendly, efficient, reliable and durable. In Indian states information system, there are many operations, which when done manually takes a lot of time and causes many errors. Due to these reasons, large numbers of problems occur. To solve above problem, and further maintain entire details of patient, doctors, and treatment etc...

Thus, after writing the code for this project, the basic concepts of object oriented program has been cleared up to some extent. Idea regarding Constructor, Method, Method Overloading, Method Overriding, static and non-static methods and variables, static keyword, this keyword and so on has been gained and implemented practically in the process of development of mentioned system.

**REFERENCES:**

a) Budd, T. (1997b), An Introduction to Object Oriented Programming, 2nd edition, Addison-Wesley

b)Ghezzi, C.,Jayazeri, M. &Mandrioli, D. (1998), Fundamentals of Software Engineering, 2ndedition, Prentice-Hall