

Dissertation Report on
HOSPITAL MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirement for the degree
of

BACHELOR OF ENGINEERING
in
COMPUTER ENGINEERING
by

Siddharth Arora 03
Zahra Bandookwala 04
Juzer Bhinderwala 08

Under the guidance of
Ms. SAIQA KHAN
(LECTURER, COMPUTER ENGINEERING DEPARTMENT,
MHSSCOE)



Computer Engineering Department
M.H. Saboo Siddik College of Engineering
University of Mumbai

TABLE OF CONTENTS

Sr. No.	Topic	Page No.
1.	Introduction.....	1
	1.1 Problem Statement.....	
2.	Design Considerations.....	
	2.1 Entity Relationship Model.....	
	2.2 Relational Model.....	
3.	Implementation.....	
	3.1 JAVA Programming Language.....	
	3.2 Oracle Database Software.....	
4.	Results and Discussion.....	
5.	Conclusion.....	
6.	Future Work.....	
7.	References	
8.	Acknowledgements	

1. **INTRODUCTION**

Hospitals help treat people with various health issues. With the advancement in technology there is now treatment for diseases that wasn't possible in earlier days, like cancer and TB.

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

JSZ's Hospital Management System is powerful, flexible, easy to use and is designed and developed to deliver real conceivable benefits to hospitals and clinics. More importantly it is backed by reliable and dependable Medical support.

This Hospital Management System 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.

1.1 PROBLEM STATEMENT

Consider a hospital management database in which the physicians and employees maintain medical and general records of patients in the hospital database.

The data requirements are as follows:

- Every patient registering will have patient no, first name, last name, blood group, address, age and sex .Patient consults a registered physician and on consultation physician admits the patient.
- Every physician treating an admitted patient is identified by their corresponding physician id, first name, last name, address and phone no.
- Every patient admitted in the hospital is only by a registered physician. It will have corresponding inpatient id, admission date, discharge date, physician admitted by, room no and bed no.
- The physician assigned also conducts the treatment.
- Every patient admitted in the hospital is treated with treatment which has respective patient no, treatment record, description, treatment name, treatment cost, physician treated by, start date, end date and result.
- After the treatment is complete patient is discharged having a receipt no, first name, last name, treatment charges, no of days admitted, and medical charges.
- The hospital consists of several departments having department no, department name and manager.
- The department's consists of employees which have employee id, first name, last name, job type, salary, resp department no and address.
- Employee's are assigned to admitted patient's for assistance and care.

2. DESIGN CONSIDERATIONS

2.1 Entity Relationship Model

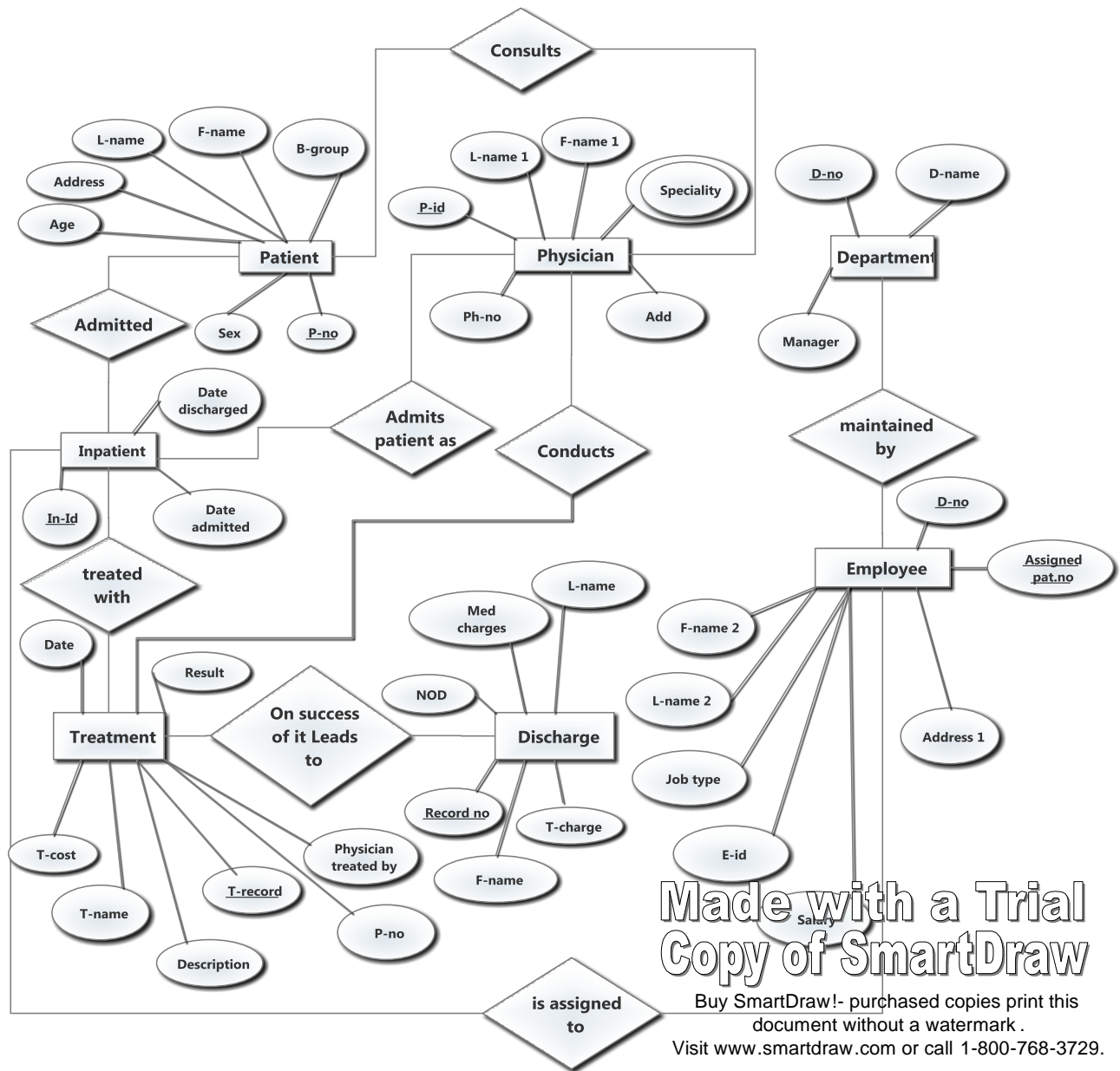


Fig.2.1 Entity Relational model of a Hospital Management

2.2 Relational Model

1]Patient

Blood_grp	Fname	Lname	Addr	age	sex	<u>Patient-no</u>
-----------	-------	-------	------	-----	-----	-------------------

2]Physician

<u>phyid</u>	Fname	Lname	Addr	ph-no	speciality
--------------	-------	-------	------	-------	------------

3]Inpatient

<u>inid</u>	date-admtd	date-dschr	<u>phy-admtdby</u>	room-no	bed-no
-------------	------------	------------	--------------------	---------	--------

4]Department

<u>deptno</u>	dname	manager
---------------	-------	---------

5]Employee

empid	Fname	Lname	Addr	job-type	salary	deptno	assngd-patno
-------	-------	-------	------	----------	--------	--------	--------------

6]Treatment

<u>trecord</u>	<u>phtytrby</u>	<u>pat-no</u>	descr	cost	sdate	result
----------------	-----------------	---------------	-------	------	-------	--------

7]Discharge

<u>rno</u>	Fname	Lname	charges	nod	medicalcharge
------------	-------	-------	---------	-----	---------------

3. IMPLEMENTATION

3.1 JAVA PROGRAMMING LANGUAGE

Swing is a graphical user interface library for the Java SE platform. It is possible to specify a different look and feel through the pluggable look and feel system of Swing. Clones of Windows, GTK+ and Motif are supplied by Sun. Apple also provides an Aqua look and feel for Mac OS X. Where prior implementations of these looks and feels may have been considered lacking, Swing in Java SE 6 addresses this problem by using more native GUI widget drawing routines of the underlying platforms.

This example Swing application creates a single window with "Hello, world!" inside:

```
// Hello.java (Java SE 5)
import javax.swing.*;

public class Hello extends JFrame {
    public Hello() {
        super("hello");
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        add(new JLabel("Hello, world!"));
        pack();
    }

    public static void main(String[] args) {
        new Hello().setVisible(true);
    }
}
```

The first import includes all of the public classes and interfaces from the javax.swing package.

The Hello class extends the JFrame class; the JFrame class implements a window with a title bar and a close control.

The Hello() constructor initializes the frame by first calling the superclass constructor, passing the parameter "hello", which is used as the window's title. It then calls the setDefaultCloseOperation(int) method inherited from JFrame to set the default operation when the close control on the title bar is selected to WindowConstants.EXIT_ON_CLOSE — this causes the JFrame to be disposed of when the frame is closed (as opposed to merely hidden), which allows the JVM to exit and the program to terminate. Next,

a JLabel is created for the string "Hello, world!" and the add(Component) method inherited from the Container superclass is called to add the label to the frame. The pack() method inherited from the Window superclass is called to size the window and lay out its contents.

The main() method is called by the JVM when the program starts. It instantiates a new Hello frame and causes it to be displayed by calling the setVisible(boolean) method inherited from the Component superclass with the boolean parameter true. Once the frame is displayed, exiting the main method does not cause the program to terminate because the AWT event dispatching thread remains active until all of the Swing top-level windows have been disposed.

3.2 ORACLE DATABASE SOFTWARE

The JDBC driver now supports large objects, including the BLOB and CLOB datatypes. This support conforms to features in the JDBC 2.0 specification, and is compatible with Oracle8i's JDBC implementation. JDBC is an application programmer's interface for accessing relational databases from Java programs. Oracle Lite supplies a native JDBC driver that allows Java applications to communicate directly with Oracle Lite's object-relational database engine. Oracle Lite's implementation of JDBC complies with JDBC 1.22. In addition, Oracle Lite provides certain extensions specified by JDBC 2.0. Oracle Lite's extensions are compatible with the Oracle8i JDBC implementation. For a complete JDBC reference, see the Sun Microsystems Web site.

To access an Oracle Lite database, your Java program must first load the Oracle Lite JDBC driver and then establish a connection to the database.

Loading the Oracle JDBC driver

To load the Oracle Lite JDBC driver, pass the fully qualified name of the Oracle Lite JDBC driver class to the Class.forName method:

```
Class.forName("oracle.lite.poljdbc.POLJDBCdriver");
```

The Class.forName method finds, loads, and links the referenced Oracle Lite JDBC driver class. You do not need to explicitly create an instance of the POLJDBCdriver class. An instance of the POLJDBCdriver class is automatically created and registered with the JDBC driver manager when you call Class.forName.

Connecting to an Oracle Database

The JDBC connect string for the Oracle Lite database is:

```
jdbc:polite:data_sourcename
```

The *data_sourcename* is the name of the data source defined in `odbc.ini`. The JDBC driver looks up the physical location of the database identified by *data_sourcename* in `odbc.ini`.

The following statement uses the JDBC connect string to connect to an Oracle Lite database with the data source name POLITE:

```
Connection conn = DriverManager.getConnection("jdbc:polite:POLITE",  
    "system", "passwd");
```

When connecting as the user system, you can provide any combination of characters for the password. However, the password cannot be blank. If the database is encrypted, you must supply the correct password.

You use the Connection object returned by `getConnection` to access and manipulate data in the Oracle Lite database. .

Using the ExecuteQuery Method

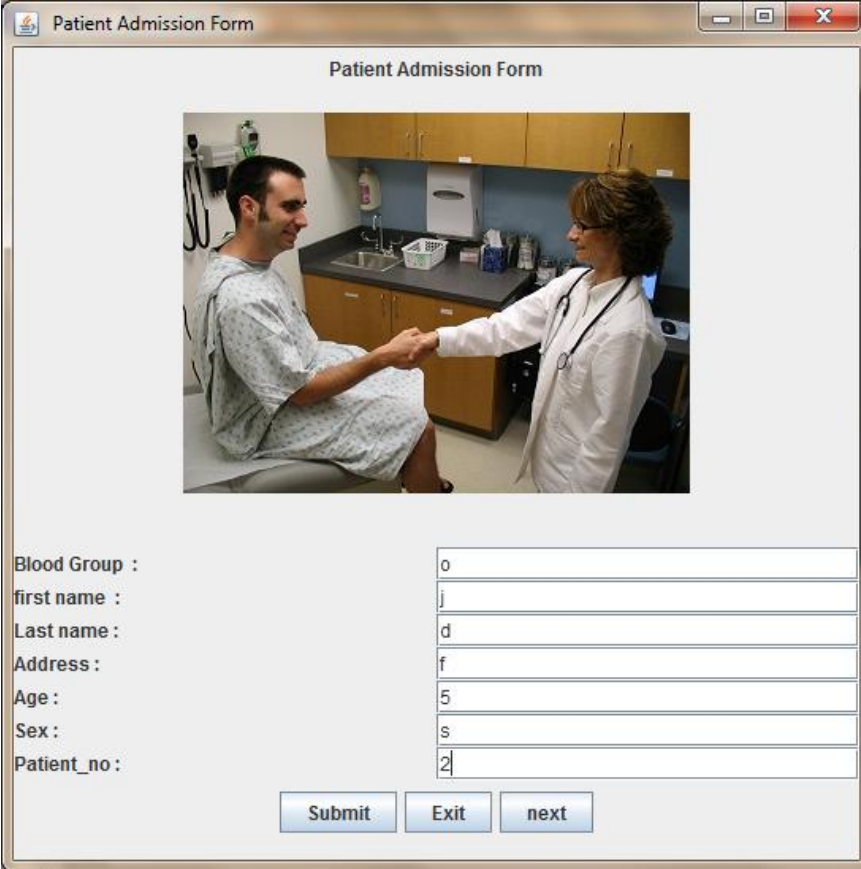
To call a stored procedure using the `executeQuery` method, first create a Statement object, which you assign the value returned by the `createStatement` method of the current connection object. You then execute the `Statement.executeQuery` method, passing it the SQL SELECT string that invokes the Java stored procedure

SQL Implementation

1. create table Patient(Blood_grp varchar(10), Fname varchar(20),Lname varchar(20),Addr varchar(50),age int,sex varchar(1),Patient_no int primary key)
2. create table Physician(phyid int primary key,Fname varchar(10),Lname varchar(10),Addr varchar(50),ph_no int,speciality varchar(20))
3. create table Inpatient(inid int primary key, foreign key(inid) references Patient(patient_no),date_admtd varchar(10),date_dschr varchar(10),phy_admtdby int,foreign key(phy_admtdby) references physician(phyid),room_no int,bed_no int)
4. create table Department(deptno int primary key, dname varchar(20),manager varchar(15))
5. create table Employee(empid int primary key,Fname varchar(10),Lname varchar(10),Addr varchar(50),job_type varchar(20),salary int,deptno int,foreign key(deptno) references department(deptno),assngd_patno int,foreign key(assngd_patno) references Inpatient(inid))
6. create table Treatment(trecord int primary key,phytrby int,pat_no int,foreign key(pat_no) references inpatient(inid),descr varchar(20),cost int,sdate varchar(10),result varchar(10))
7. create table Discharge(Rno int primary key, foreign key(rno) references Treatment(trecord),Fname varchar(10),Lname varchar(10),charges int,nod int,medicalcharge int)

4. RESULTS AND DISCUSSION

This section deals with the snapshots of the result of our project



The screenshot shows a window titled "Patient Admission Form". Inside the window, there is a photograph of a male patient in a hospital gown sitting on an examination table, shaking hands with a female healthcare professional in a white lab coat. Below the photo, there are input fields for patient information:

Field	Value
Blood Group :	o
first name :	j
Last name :	d
Address :	f
Age :	5
Sex :	s
Patient_no :	2

At the bottom of the form, there are three buttons: "Submit", "Exit", and "next".

Fig.4.1 Patient Registration form

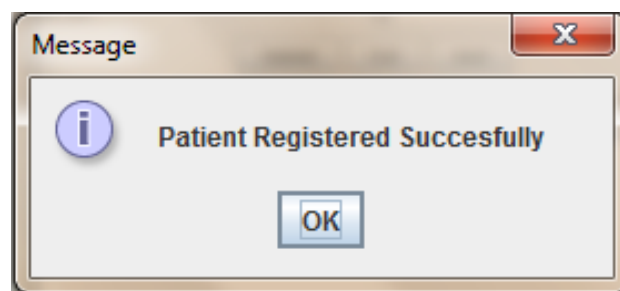
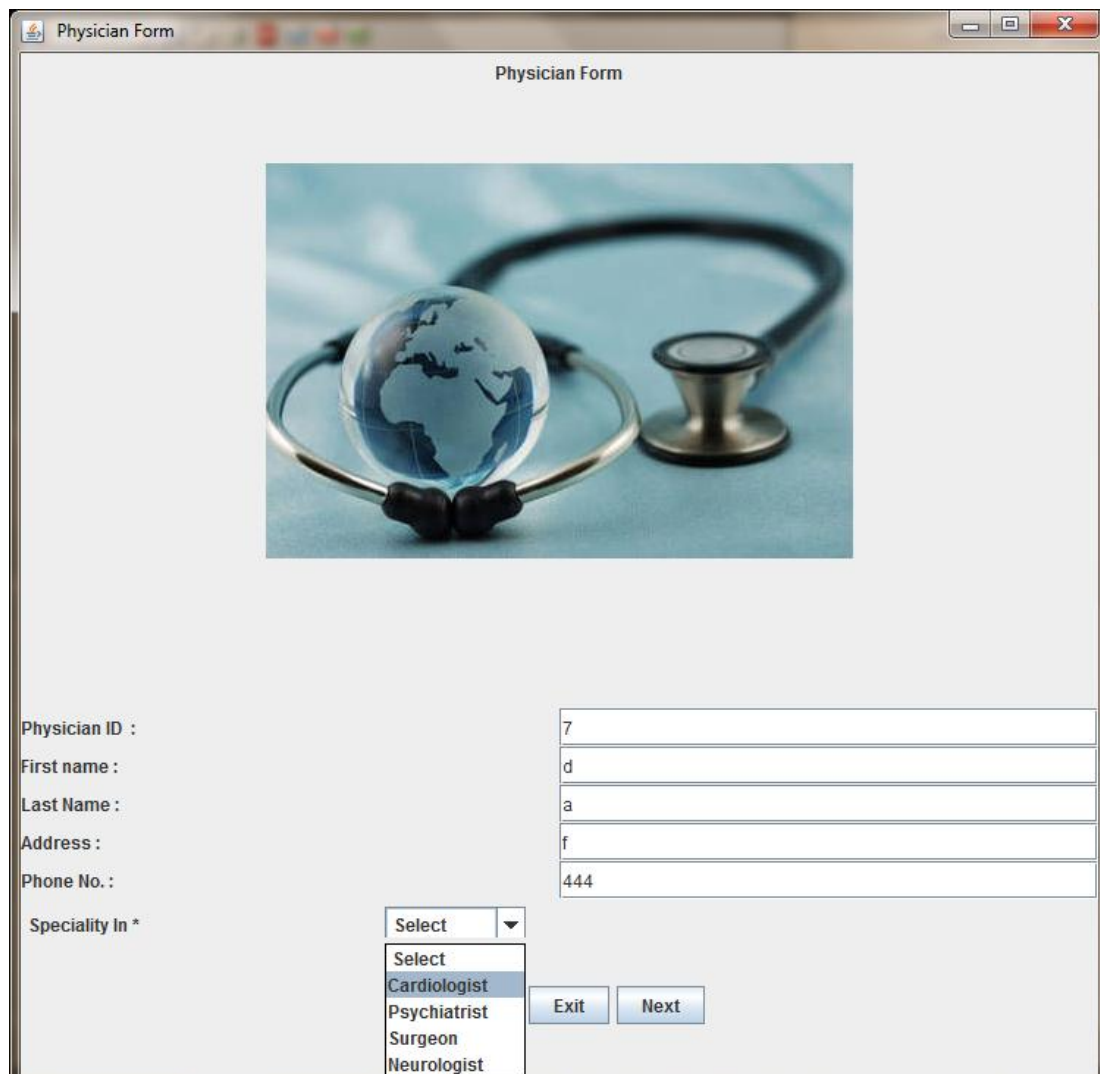


Fig.4.2 Successfully Registered



The image shows a software window titled "Physician Form". At the top center is a photograph of a stethoscope with a small globe of the Earth as its chest piece, resting on a light blue surface. Below the image are several input fields and a dropdown menu. The labels on the left are: "Physician ID :", "First name :", "Last Name :", "Address :", "Phone No. :", and "Speciality In *". The corresponding input fields on the right contain the following data: "7", "d", "a", "f", and "444". The "Speciality In *" field is a dropdown menu with a "Select" button and a downward arrow. The menu is open, showing the following options: "Select", "Cardiologist", "Psychiatrist", "Surgeon", and "Neurologist". At the bottom right of the form are two buttons: "Exit" and "Next".

Field	Value
Physician ID	7
First name	d
Last Name	a
Address	f
Phone No.	444
Speciality In *	Cardiologist

Fig.4.3 Physician form

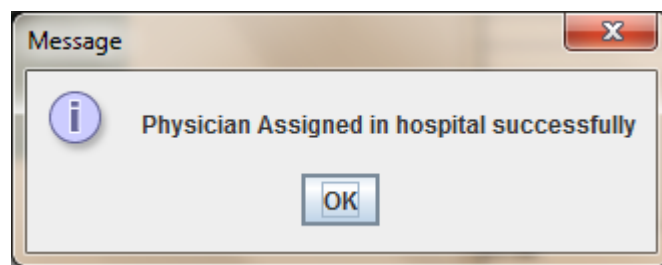


Fig.4.4 Successfully Assigned Physician

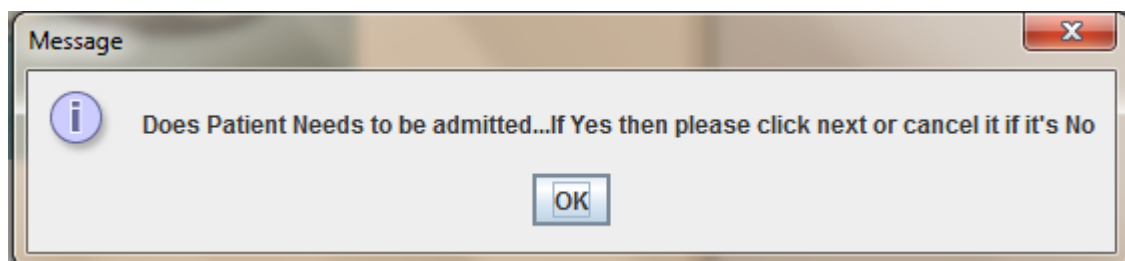


Fig.4.4 Request to admit the patient or not

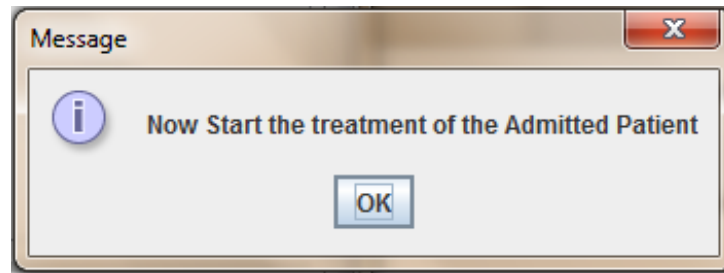


Fig.4.5 Confirmation of all details will help start the treatment of admitted patient

A screenshot of a 'Treatment Form' window. The title bar says 'Treatment Form' with standard window controls. The main area contains a photograph of a woman lying in a hospital bed, wearing a hospital gown and having an IV line. Below the photo, there is a form with the following labels and values:

treatment record :	5
physician treated by :	4
patient no :	5
description :	cancer
cost :	100000
Start date :	12/03/05
Result :	Death

At the bottom of the form are three buttons: 'Submit', 'Exit', and 'Discharge'.

Fig.4.6 Treatment Record form

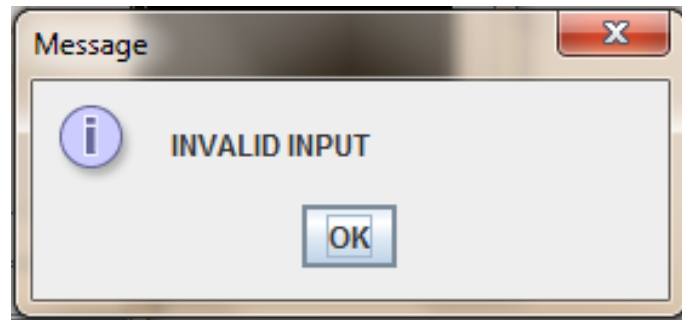


Fig.4.6.1 Message Displayed due to unmatched identity from the record

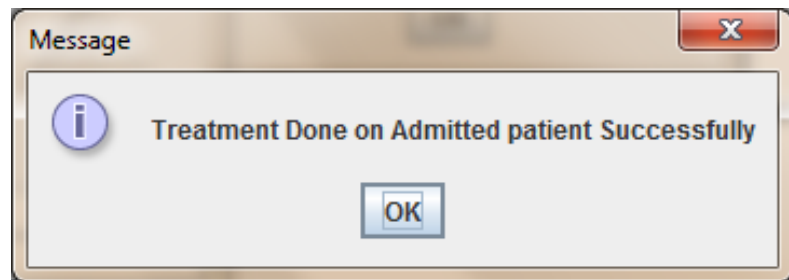


Fig.4.6.2 Message Displayed on accord with the matched identity from the record

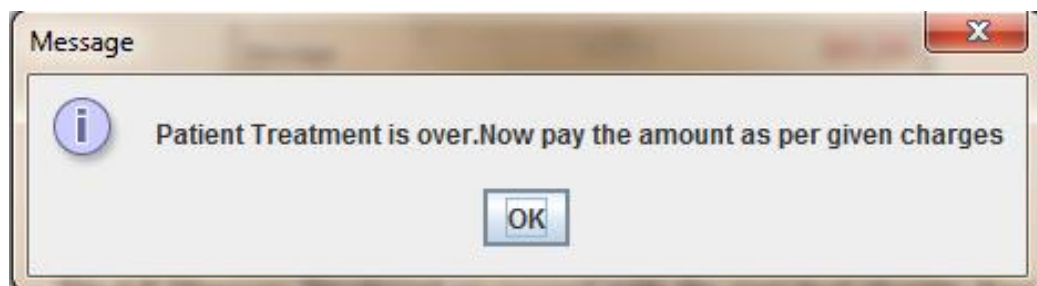


Fig.4.6.3 Completion of treatment leads to discharge

DISCHARGE FORM

Discharge form

Ward Charges = Rs.1000/day

receipt no : 5

first name : sfd

Last name : sdf

treatment charges : 100000

No of days : 5

Medical charges : 70000

Submit Exit

Fig.4.7 Discharge Form

Message

i AMOUNT TO BE PAID :175000

OK

Fig.4.7.1 Total Amount Generated

5. FUTURE WORK

In this project we learnt all the basics of database management but due to lack of time and some knowledge we could not implement all that we had planned. So, in our future work we will include the following works :-

- We will include various tables for each required entity.
- We will include various mode of payment.
- We have to use CLOB to store images in our database.
- We wanted to generate the amount of total treatment by calculating the no. of days the treatment was going on.
- We will create complete history of a Patient from its registration right till the date of discharged from hospital.

6. CONCLUSION

- We have successfully completed our database project on Hospital Management System.
- From this project we learnt how to create a database along with its connectivity.
- We understood the concepts of the various classes present within javaswing.
- We used Applets to make our frames more presentable and learnt how to create tables and insert values in oracle using java.
- We used different ways to display the tables.

ACKNOWLEDGEMENT

To God, for His loving guidance and for the many blessings He has bestowed upon us.

“Become a student of change. It is the only thing that will remain constant.” These words of our beloved teacher **ER. SAIQA KHAN helped** us to encourage ourselves to tackle all the difficulties in our project plus with her guidance for this project & also for her constant encouragement, valuable help in assisting us in every possible ways for completion of this report and project. Her helping hand has been instrumental in our achievements. She has also provided a calming influence over the course of this hectic schedule and helped us remain in control over the entire proceedings. Our foremost thanks go to our project partners and help of our well-wishers and colleagues.

We are grateful to all the staff members, non-teaching staff & all our friends for giving us the helping hand and letting us share the resources which were related to our project work

REFERENCES

1.	Herbert Schildt, "Java The Complete Reference", Fifth Edition,
2.	Navathe and Elmasri, Fundamental of Database System,Fourth Edition
3.	Avi Silberschatz, Henry F. Korth,S Sudarshan,Database system Concepts,Fifth Edition
4.	"java swing"[online]:Available www.sun.java.com
5.	JDBC notes by Er Saiqa Khan