**Submitted by**

***Ashish Kumar Verma***

***Gagan Kumar***

***Sunil Kumar***

***Mohammad Hatim***

**Under the guidance of**

Mr. Akhilesh Kumar

**APTECH COMPUTER EDUCATION**

**81, DEFENCE ENCLAVE**

**PREET VIHAR**

**DELHI**

**CERTIFICATE**

This is to certify that this is a bonafied record of the project work done satisfactorily at **Aptech Computer Education** By **Mr. *Ashish Kumar Verma, Gagan Kumar, Sunil Kumar, Mohammad Hatim*** has been carried out under my direct supervision and guidance. This report or similar report on the topic has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

**Signature of Guide/Supervisor** **Signature of Candidate**

(AKHILESH KUMAR ) ***Ashish Kumar Verma***

***Gagan Kumar***

***Sunil Kumar***

***Mohammad Hatim***

**Signature:**

**PLACE: DELHI** **Name: Mr. DIVYA KAPOOR**

**Designation:** **Centre Head**

**DATE: 20-11-2009**

**APTECH COMPUTER EDUCATION**

**PREET VIHAR CENTRE**

**DELHI**

**ACKNOWLEDGEMENT**

I thankfully acknowledge the encouragement and guidance by **Mr. AKHILESH KUMAR**  without whose help report could not have achieved its present form. His continual attention and encouragement helped keep me going.

I am highly indebted to **APTECH COMPUTER EDUCATION** for providing excellent support for the project.

**(*Ashish Kumar Verma)***

***(Gagan Kumar)***

***(Sunil Kumar)***

***( Mohammad Hatim)***

**TABLE OF CONTENTS**

**SNO. TOPIC PAGENO.**

**PART – I**

1. **DOCUMENTATION………………………………………………………..6**

**-PROBLEM DEFINITION…….……………………………...6**

**-REQUIREMENT SPECIFICATION.……………………….7**

**-BACKGROUND OF THE PROJECT……………………….7**

**-ANALYSIS ( Activity Chart) .………………………………...8**

**-METHODOLOGIES USED…………………………………..9**

**-HARDWARE CONFIGURATION OF THE SYSTEM…...13**

**-SOFTWARE REQUIREMENTS…………………..…….….13**

**-SOFTWARE MAINTENANCE & EVALUATION..………14**

**-DIAGRAMATIC REPRESENTATION OF THE S/W…….15**

#### 

**PART – II**

1. **OUTPUT SCREENS…………….…………………………………………22**

PART – I

**DOCUMENTATION**

**Problem Definition**

'SANJEEVANI' is a hospital providing the general health services. The information about the patients admitted in the hospital is maintained properly yet such information is not always easily accessible.

There is growing realization that the more widespread use of information technology to increase efficiency and enable changes in health care delivery processes could do much to improve the performance of the health care delivery processes could do much to improve the performance of the health care system, within the bounds of appropriate measures to protect the confidentiality of private health information . Hence the higher authorities of hospital have dicided to for a computer based patient record system includes all the elements that facilitate the capture , storage, processing , communication , security , and presentation of computer based patient record information.

**Requirements Specification**

Depending on the decision given by the management on the Computer Based Patient Information system is to be developed. Following are the requirements based on which the system needs to be developed.

1. The system will support registering patients

2. Users of this system can search for patient by name , admission date, discharge date etc.

3. Users can view the previous visit histories of any patient.

4. System can maintain the list of doctor in the hospital.

5. System can maintain the list of beds/rooms available in the hospital.

6. Patietnt are categorezed into "In Patient" and "Out Patients"

7. Perform necessary validations.

**BACKGROUND OF THE PROJECT**

**OOPS CONCEPT**

**JAVA** is an object oriented programming language. It was developed by JAMES GOSLING, PATRIC NAUGHTON, CRIS WARTH, MIKE SHERIDAN and 1985. They[ wanted to combine the best of both the languages and create a more powerful language that could support object oriented programming language.

The main problem associated with large programs written in procedural language is under evaluation if data. Data is neglected. There is no built in mechanism to ensure the security of data. The OOP approach is different. In OOP, the emphasis is on data not on procedures. In OOP, a class is the specification for such a data form.

In OOP, we bundle together the data and the functions that operate on the data into a single software unit called class. A class is the specification of a data entity. This data entity is called object. An object is an instantiation of a class. The relationship between a class and an object is the same as that of a built in data type and a variable of that type. A class is a data type and an object is an instance of that data type.

The most important and the most basic concept in OOP is the concept of the class. Define a class does not perform any memory allocation a class is like a blue print for a house. The blue print gives the specification of the house. Based on the blue print many houses can be constructed. In a similar way you can create many objects of a given class. When an object is created, memory allocation takes place. OOP design methodology is different. Problems are no longer divided into functions. Instead it is divided into a collection of mutually interacting objects. Objects interact with one another through member functions.

DATA HIDING

The wrapping up of the data and functions into a single unit (called classes ) is know as encapsulation. Data encapsulation is the most striking feature of a class. The data is not accessible to the outside world, and only those functions, which are wrapped in the class program. This insulation of data from direct access by the program is called data hiding or information hiding.

FUNCTIONALITY

In the making of this software I used basically modularity or functionality in coding part of this software. I used so many functions in making of this software and performing different types of task, which is easy to me, do my work with great efficiency. In modularity of this software JAVA supports very well. I divide the different task in different modules or functions, where every module is performing its individual task and give the required result.

**Analysis:-**

#### System Planning

1. Activity Chart

**Activity chart:-**

Patient

Entry

Attendant

Patient Admit

Patient Discharge

Payment

Status

Report

f

#### Methodologies Used:-

#### Programming Language(JAVA)

B. Database(Back-End)

**Programming Language(JAVA):**

JAVA is the most widely used object-oriented language today. It is faster and more powerful than Java, another popular object-oriented language, which lacks certain features such as pointers and multiple inheritance.

Some important concepts of the object-oriented programming language are as follows:

* Objects
* Classes
* Data abstraction and encapsulation
* Inheritance
* Polymorphism
* Dynamic binding
* Message passing

We shall discuss these concepts in detail in this Section.

Objects:

Object are the basic run-time entities in an object-oriented system. They may represent a person, a bank account, a table of data or any item that the program has to handle. They may also represent user-defined data such as vectors, time and lists. Programming problem is analyzed in terms of objects and the nature of communication between them. Program objects should be chosen such that they match closely with the real-world objects. Objects take up space in the memory and have an associated address like a record in Pascal, or a structure in C.

**Classes:**

We just mentioned that objects contain data, and code to manipulate that data. The entire set of data and code of an object can be made a user-defined data type with the help of a class. In fact, objects are variables of the type class. Once a class has been defined, we can create any number of objects belonging to that class. Each object is associated with the data of type class with which they are created. A class is thus a collection of objects of similar type. Classes are user-defined data types and behave like the built-in types of a programming language. The syntax used to create an object is no different than the syntax used to create an integer object in C.

**Data Abstraction and Encapsulation:**

The wrapping up of data and functions into a single unit (called class ) is know as encapsulation. Data encapsulation is the most striking feature of a class. The data is not accessible to the outside world, and those functions which are wrapped in the class can access it .These functions provide the interface between the object’s data and the program. This insulation of the data from direct access by the program is called ***data hiding***or ***information hiding****.*

Abstraction refers to the act of representing essential features without including the background details or explanations. Classes use the concept of abstraction and are defined as a list of abstract *attributes* such as size, weight and cost, and functions to operate on these attributes. They encapsulate all the essential properties of the object that are to be created. The attributes are sometimes called data members because they hold information. The functions that operate on these data are sometimes called methods or *member functions.*

Since the classes use the concept of data abstraction, they known as Abstract Data Types (ADT).

**Inheritance:**

Inheritance is the process by which objects of one class acquire the properties of objects of another class. It supports the concept of hierarchical classification. For example, the bird ‘robin’ is a part of the class ‘flying bird’ which is again a part of the class ‘bird’ . The principle behind this sort of that each derived class shares common characteristics with the class from which it is derived.

#### In OOP, the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a new class from the existing one. The new class will have the combined features of both the classes. The real appeal and power of the inheritance mechanism is that it allows the programmer to reuse a class that is almost, but not exactly, what he wants, and to tailor the class in such a way that it does not introduce any undesirable side-effects into the rest of the classes.

Note that each sun-class defines only those features that are unique to it. Without the use of classification, each class would have to explicitly include all of its features

.

**Polymorphism:**

Polymorphism is another important OOP concept. Polymorphism, a Greek term, means the ability to take more than one form. An operation may exhibit different behaviors depends upon the types of data used in the operation. For example, consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strings, then the operation would produce a third string by concatenation. The process of making an operator to exhibit different behaviors in different instances is know as *operator overloading.*

A single function name can be used to handle different number and different types of arguments. This is something similar to a particular word having several different meanings depending on the context. Using a single function name to perform different types of tasks is known as ***function overloading****.*

Polymorphism plays an important role in allowing objects having different internal structures to share the same external interface. This means that a general class of operations may be accessed in the same manner even though specific action associated with each operation may differ. Polymorphism is extensively used in implementing inheritance.

**Dynamic Binding:**

Binding refers to the linking of a procedure call to the code to be executed in

response to the call. Dynamic binding (also known as late binding ) means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance. A function call associated with a polymorphism with a polymorphism reference depends on the dynamic type of that reference.

By inheritance, every object will have this procedure. Its algorithm is, however, unique to each object and so the draw procedure will be redefined in each class that defines the object. At run-time, the code matching the object under current reference will be called.

**Message Passing**

An object-oriented program consists of a set of objects that communicate with each other. The process of programming in an object-oriented language, therefore, involves the following basic steps:

* 1. Creating classes that define objects and their behaviors,
  2. Creating objects from class definitions, and
  3. Establishing communication among objects.

Objects communication with one another by sending and receiving information much the same way as people pass messages to one another. The receiving information much the same way as people pass message to one another. The concept of message passing makes it easier to talk about building systems that directly model or simulate their real-world counterparts.

A message for an object is a request for execution for execution of a procedure, and therefore will invoke a function (procedure) in the receiving object that generates the desired result. Message passing involves specifying the name of the object, the name of the function (message) and the information to be sent.

Objects have a life cycle. They can be created and destroyed. Communication with an object is feasible as long as it is alive.

**Hardware Configuration of the system**

1. A minimum computer system that will help you access all the tools in the courses is a Pentium 166 or better
2. 128 megabytes of RAM or better
3. Window 2000 server(or higher if possible)

**Software Requirements:**

**Software**

1. Java/J2EE

**Database**

1. SQL Server

#### 5.Software maintenance & Evaluation

#### Maintenance is the enigma of system development. It holds the software industry captive, tying up programming resources. Analysts and programmers spend far more time maintaining programs than do writing them. Maintenance accounts for 50-80 percent of total system development. Some programs executed with software maintenance.

#### 

#### Maintenance is not as rewarding as exciting as developing system. It is perceived as requiring neither skill nor experience.

#### Users are not fully cognizant of the maintenance problem or its high cost.

#### Few tools and techniques are available for maintenance.

#### A good test plan is lacking.

#### Standards, procedures, and guidelines are poorly defined and enforced.

#### Maintenance is viewed as a necessary evil, often delegated to junior programmers. There is practically no maintenance manager job classification in the MIS field.

#### Programs are often maintained without care for structure and documentation.

#### There are minimal standards for maintenance.

#### 9. Programmers expect that they will not be in their current commitment by the time their programs go into the maintenance cycle.

#### 6.Software Architecture

#### Diagrammatical Representation of The Software

#### Data Flow Diagram (DFD)

#### DIAGRAMMATIC

REPRESENTATION

OF

THE

SOFTWARE

##### MAIN MENU

**MAIN MENU**

Add New Patientte

Admit patienttes

Discharge

patienttes

Display

Edit

Quit

##### Patient

ADD

PATIENTT

UPDATE

LIST MENU

SEARCH CODE

EXIT

ALL PATIENT

PATIENT NAME

DOCTOR

OTHER DETAILS

DISCHARGE

**PATIENT**

ADMIT/DISCHARGE

**ADMIT/DISCHARGE**

ADMIT

DISCHARGE

LIST

PATIENT

ADMITD

RETURN TO MAIN MENU

**C. Data Flow Diagram**

**MAIN PROCESS**

**A B C**

Patient

Patient Status Report

Collection

Report

**Process**

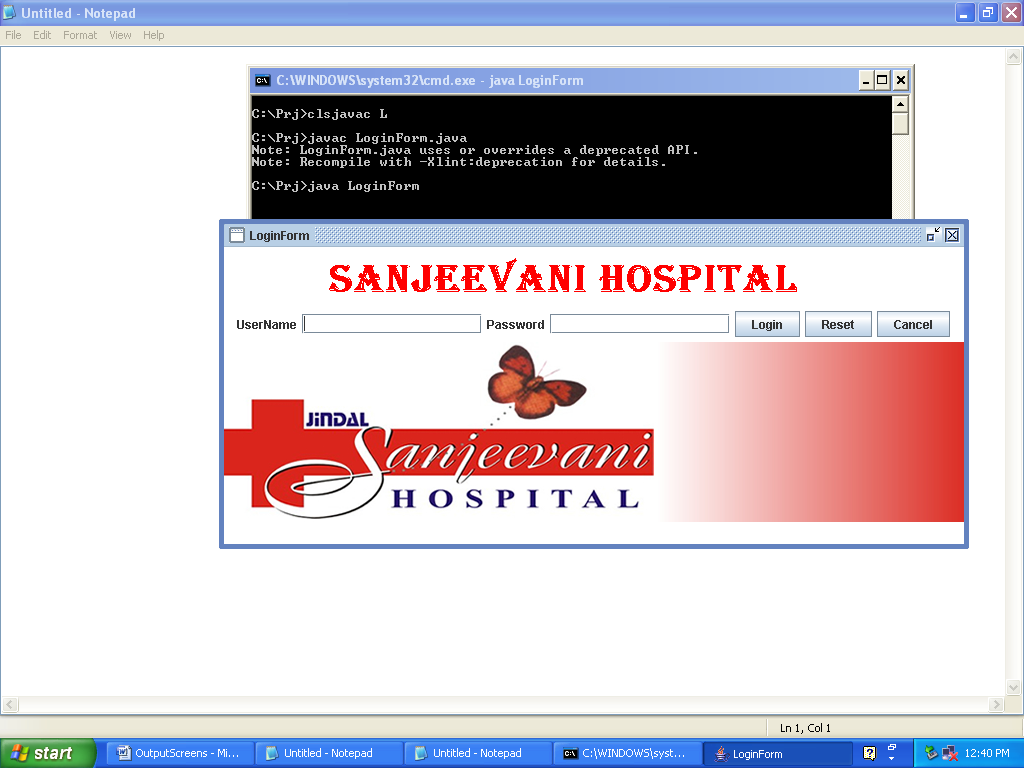
Patient Registration Number

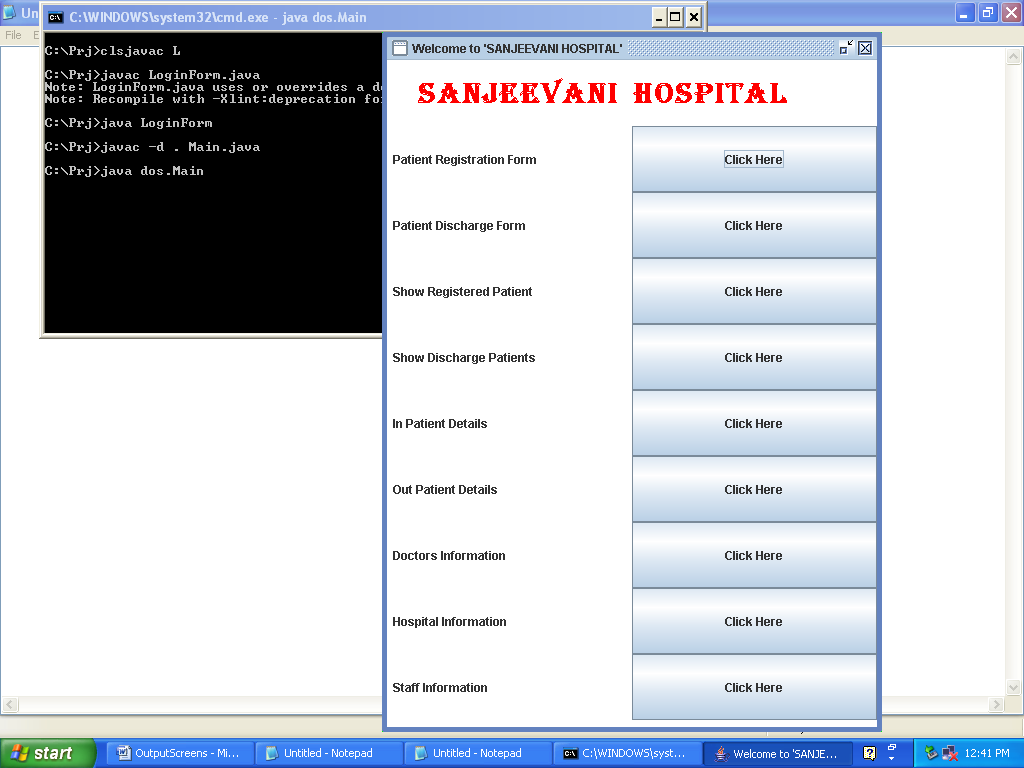
**Patient**

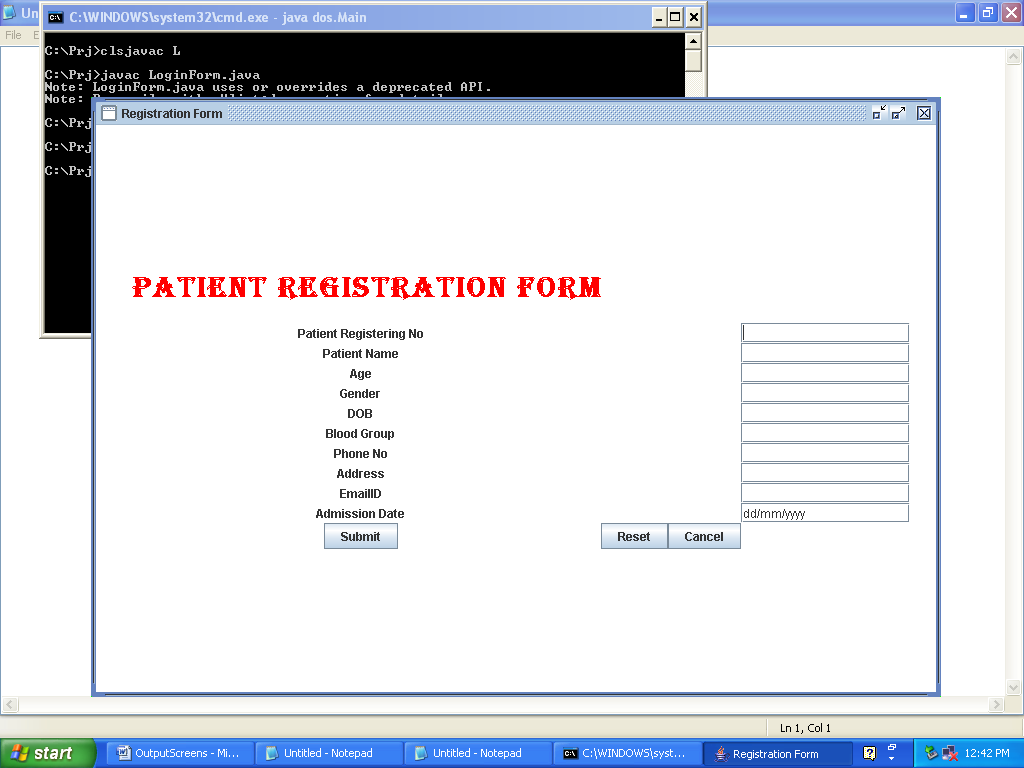
**PART-II**

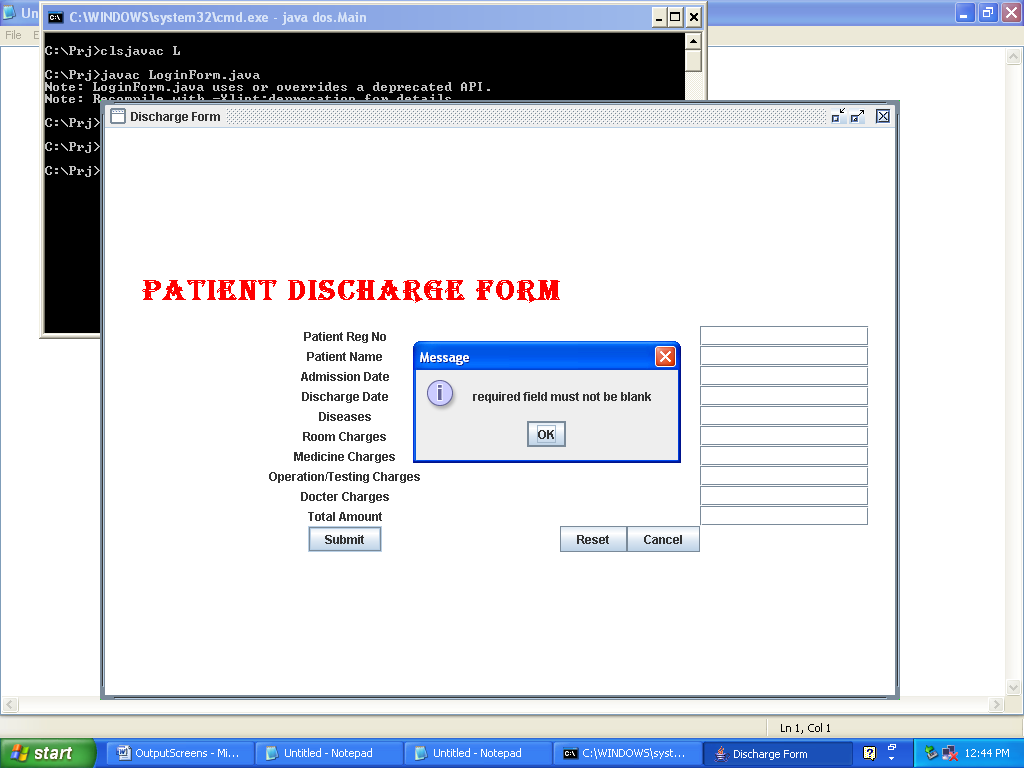
**OUTPUT**

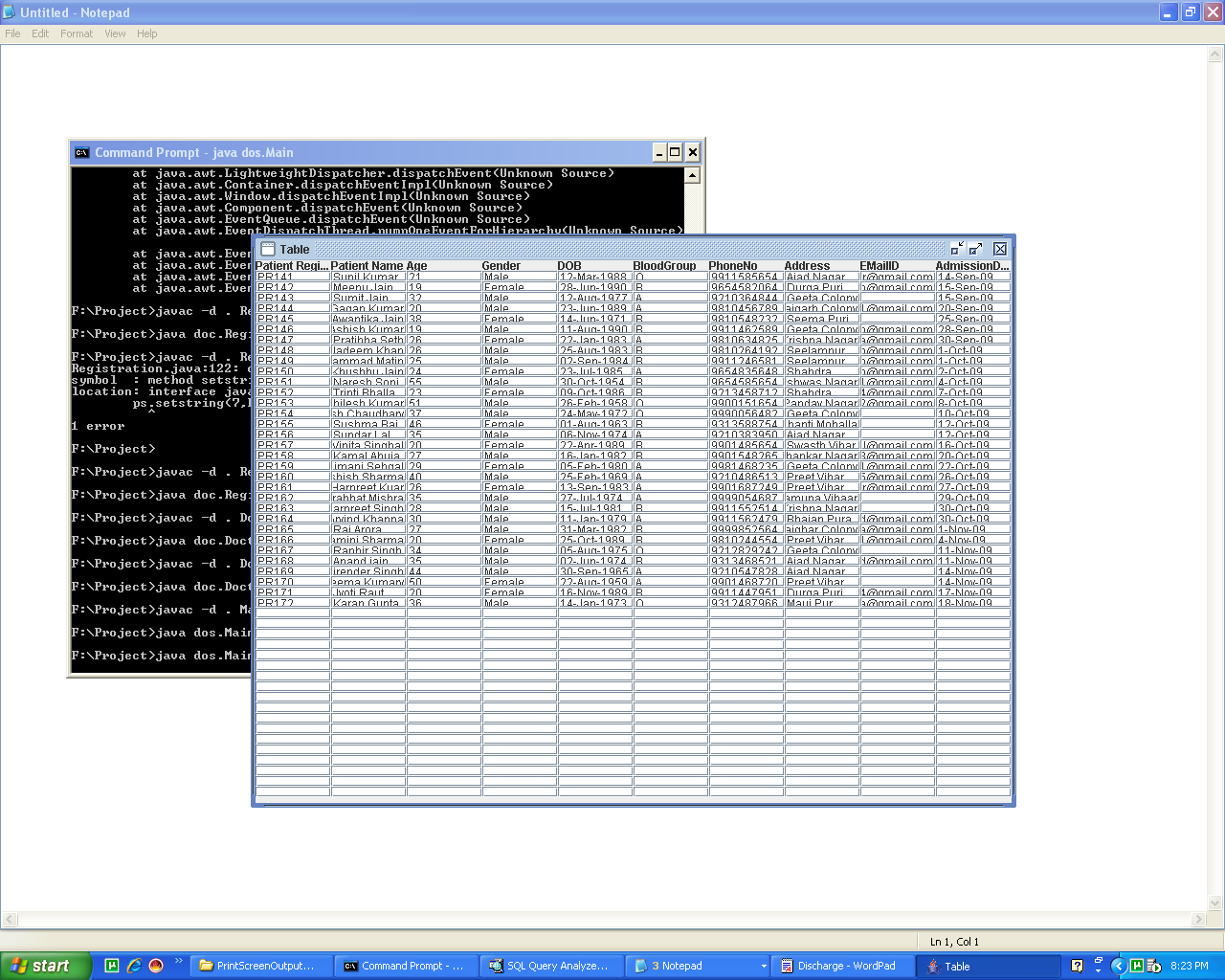
**SCREENS**

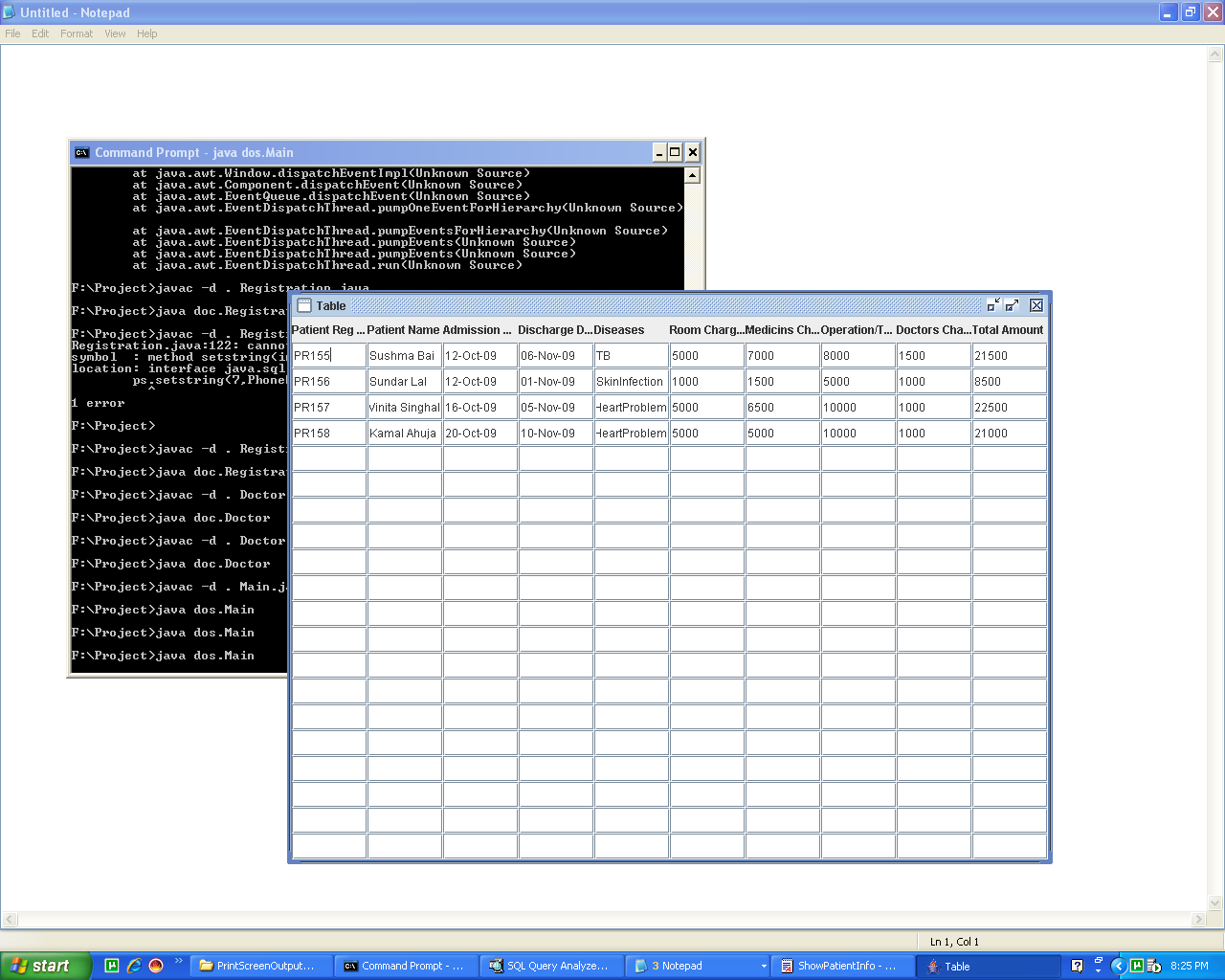


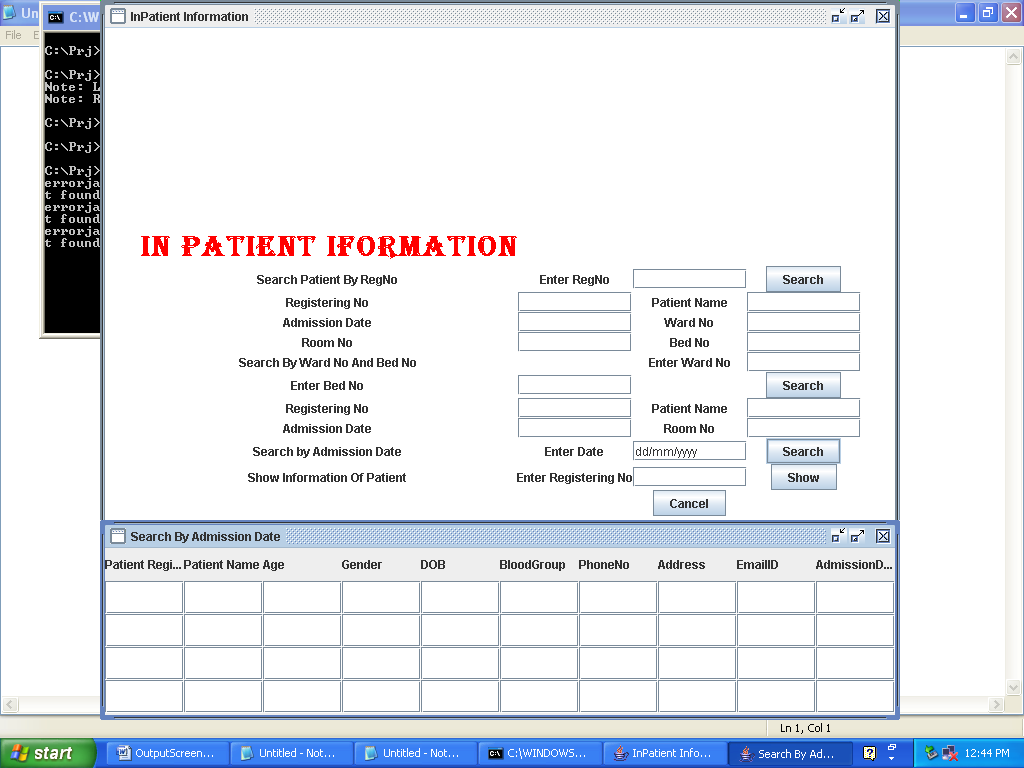


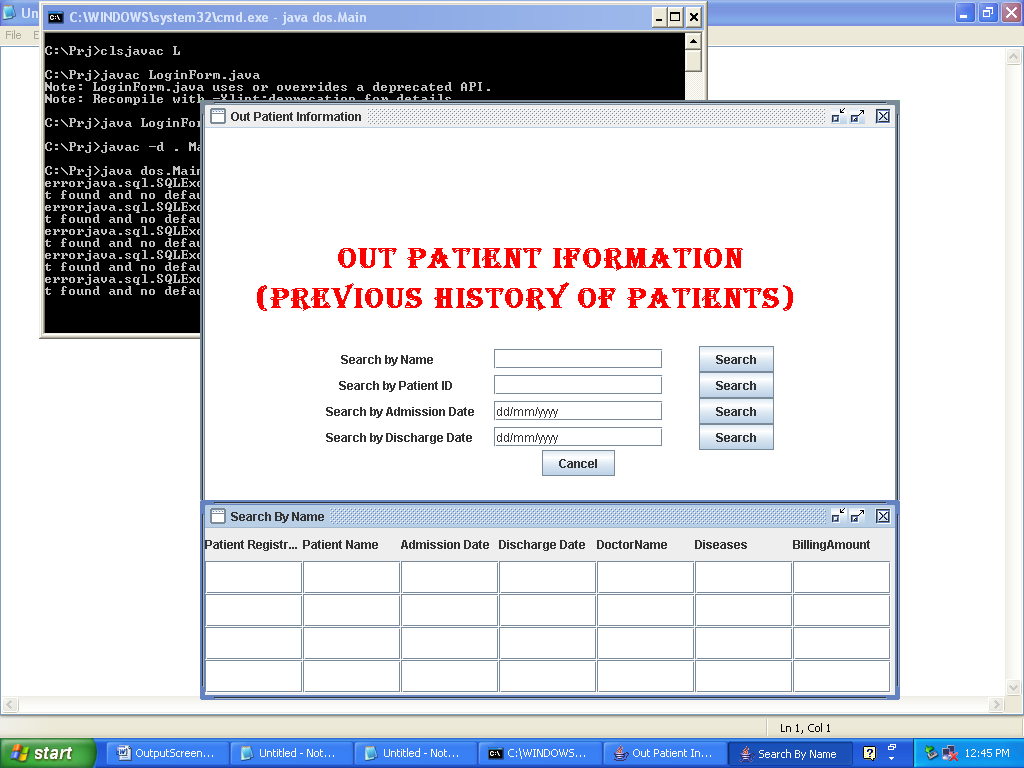


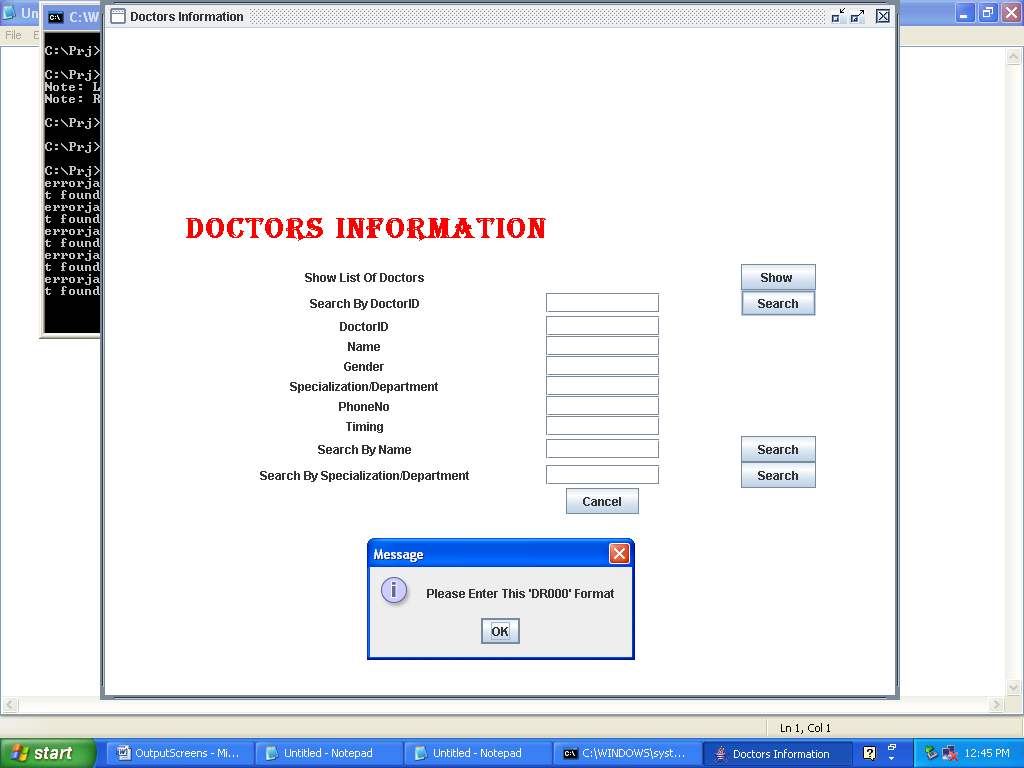


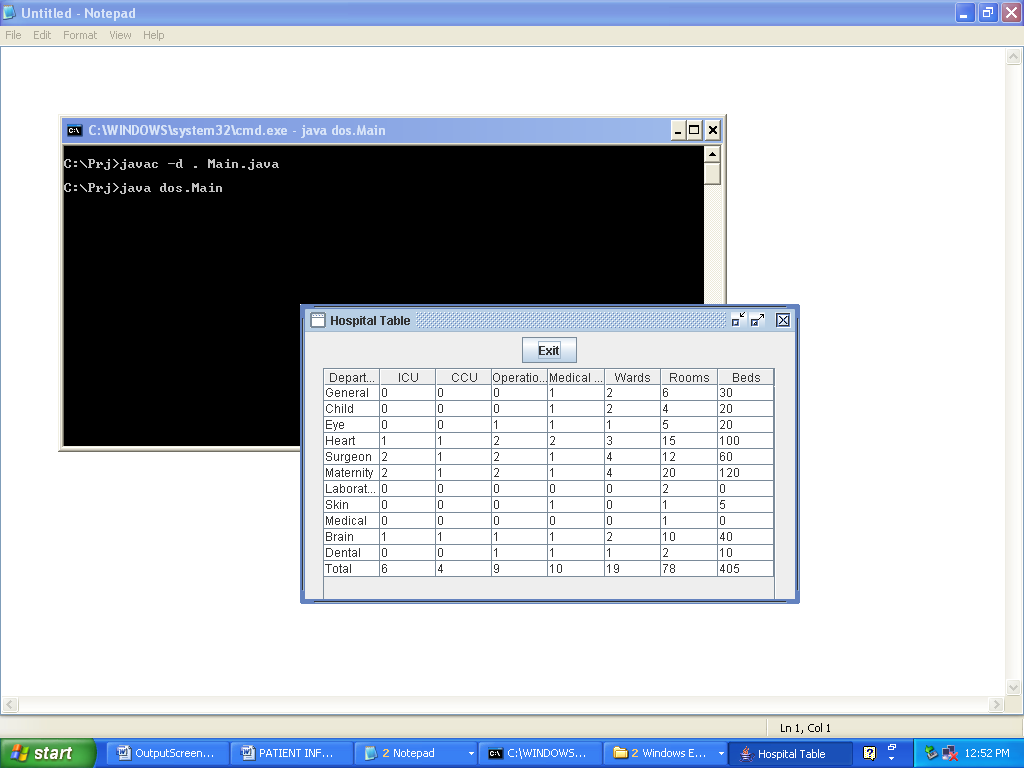










****

