Synopsis on

Hospital Management System

EXECUTIVE SUMMARY

Hospital 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. More importantly it is backed by reliable and dependable support.

The project 'Hospital Management System' is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses JAVA as the front-end software which is an Object Oriented Programming and has connectivity with MY SQL.

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Medicine details, Doctor, Wards, , Admin, Store, Patient appointment, bill payment, record modification, discharge details etc.

PROJECT

SYNOPSIS

Hospital Management System

INTRODUCTION TO THE STUDY

Introduction:

Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, part-by-part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e. Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man.

Medical Institutions in India

Still being a developing nation India has seen a tremendous growth of the Health sector in the field of research as well as in the field of development of numerous large and small scale Hospital institutions still lacking in inter-structure facilities. Government of India has still aimed at providing medical facilities by establishing hospital. The basic working of various hospitals in India is still on paper as compared to hospitals in European countries where computers have been put in to assist the hospital personals their work. The concept of automation of the administration and management of hospital is now being implemented in India also, with large hospitals like APPOLO and AIIMS in Delhi, ESCORTS in Chennai, having automated their existing system.

Our project is based on the above concept i.e. automation of Administration and Management of Hospital. The project has been developed keeping in-view the following aspects: -

(i) Working environment of the Hospital.

(ii) The thought-process and attitude of Indian people.

(iii) The literacy rate of India.

(iv) The Existing system, being used in the majority of Hospitals.

(v) The availability of Infra-structural facilities likes finance, skilled personals, and working environment.

DEFINITION OF PROBLEM:

Since HOSPITAL is associated with the lives of common people and their day-to-day routines so I decided to work on this project.

The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like Room activities, Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc.

I have tried my best to make the complicated process **Hospital Management System** as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each Hospital's activity in computerized way rather than manually which is time consuming.

I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

What is SQL Used for:

Using SQL one can create and maintain data manipulation objects such as table, views, sequence etc. These data manipulation objects will be created and stored on the server's hard disk drive, in a table space, to which the user has been assigned.
Once these data manipulation objects are created, they are used extensively in commercial applications.

DML, DCL, DDL:

In addition to the creation of data manipulation objects, the actual manipulation of data within these objects is done using SQL.

The SQL sentences that are used to create these objects are called DDL's or Data Definition Language. The SQL sentences used to manipulate data within these objects are called DML's or Data Manipulation Language. The SQL sentences, which are used to control the behavior of these objects, are called DCL's or Data Control Language.

doctor is assigned to the patient according to the patient's disease. And if any patient is getting discharged, the bed assigned to him/her should automatically free in the computer.

Control: The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

Security: Security is the main criteria for the proposed system. Since illegal access may corrupt the database and it will affect not only the hospital but also it also affects the patient's life. So security has to be given in this project.

SOFTWARE & HARDWARE REQUIREMENTS:

Adding dynamic content via expressions

As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp. Of course, what makes JSP useful is the ability to embed Java. Put the following text in a file with .jsp extension (let us call it **`hello.jsp`**), place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello!  The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Notice that each time you reload the page in the browser, it comes up with the current time.

The character sequences `<%=` and `%>` enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dynamic HTML pages that change in response to user actions or vary from user to user.

Exercise: Write a JSP to output the values returned by `System.getProperty` for various system properties such as `java.version`, `java.home`, `os.name`, `user.name`, `user.home`, `user.dir` etc.

Scriptlets

We have already seen how to embed Java expressions in JSP pages by putting them between the <%= and %> character sequences.

But it is difficult to do much programming just by putting Java expressions inside HTML.

JSP also allows you to write blocks of Java code inside the JSP. You do this by placing your Java code between <% and %> characters (just like expressions, but without the = sign at the start of the sequence.)

This block of code is known as a "scriptlet". By itself, a scriptlet doesn't contribute any HTML (though it can, as we will see down below.) A scriptlet contains Java code that is executed every time the JSP is invoked.

Here is a modified version of our JSP from previous section, adding in a scriptlet.

```
<HTML>
<BODY>
<%
    // This is a scriptlet.  Notice that the "date"
    // variable we declare here is available in the
    // embedded expression later on.
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello!  The time is now <%= date %>
</BODY>
</HTML>
```

If you run the above example, you will notice the output from the "System.out.println" on the server log. This is a convenient way to do simple debugging (some servers also have techniques of debugging the JSP in the IDE. See your server's documentation to see if it offers such a technique.)

By itself a scriptlet does not generate HTML. If a scriptlet wants to generate HTML, it can use a variable called "out". This variable does not need to be declared. It is already predefined for scriptlets, along with some other variables. The following example shows how the scriptlet can generate HTML output.

```
<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date"
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello!  The time is now
<%
    // This scriptlet generates HTML output
    out.println( String.valueOf( date ));
%>
</BODY>
</HTML>
```

Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type javax.servlet.jsp.JspWriter.

Another very useful pre-defined variable is "request". It is of type javax.servlet.http.HttpServletRequest

A "request" in server-side processing refers to the transaction between a browser and the server. When someone clicks or enters a URL, the browser sends a "request" to the server for that URL, and shows the data returned. As a part of this "request", various data is available, including the file the browser wants from the server, and if the request is coming from pressing a SUBMIT button, the information the user has entered in the form fields.

The JSP "`request`" variable is used to obtain information from the request as sent by the browser. For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.) Let us modify the code as shown:

```
<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date"
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello!  The time is now
<%
    out.println( date );
    out.println( "<BR>Your machine's address is " );
    out.println( request.getRemoteHost());
%>
</BODY>
</HTML>
```

A similar variable is "response". This can be used to affect the response being sent to the browser. For instance, you can call `response.sendRedirect( anotherUrl );` to send a response to the browser that it should load a different URL. This response will actualy go all the way to the browser. The browser will then send a different request, to "`anotherUrl`". This is a little different from some other JSP mechanisms we will come across, for including another page or forwarding the browser to another page.

Exercise: Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.

Mixing Scriptlets and HTML

We have already seen how to use the "`out`" variable to generate HTML output from within a scriptlet. For more complicated HTML, using the out variable all the time loses some of the advantages of JSP programming. It is simpler to mix scriptlets and HTML.

Suppose you have to generate a table in HTML. This is a common operation, and you may want to generate a table from a SQL table, or from the lines of a file. But to keep our example simple, we will generate a table containing the numbers from 1 to N. Not very useful, but it will show you the technique.

Here is the JSP fragment to do it:

```
<TABLE BORDER=2>
<%
    for ( int i = 0; i < n; i++ ) {
        %>
        <TR>
        <TD>Number</TD>
```

```
        <TD><%= i+1 %></TD>
        </TR>
        <%
    }
%>
</TABLE>
```

You would have to supply an `int` variable "n" before it will work, and then it will output a simple table with "n" rows.

The important things to notice are how the `%>` and `<%` characters appear in the middle of the "`for`" loop, to let you drop back into HTML and then to come back to the scriptlet.

The concepts are simple here -- as you can see, you can drop out of the scriptlets, write normal HTML, and get back into the scriptlet. Any control expressions such as a "while" or a "for" loop or an "if" expression will control the HTML also. If the HTML is inside a loop, it will be emitted once for each iteration of the loop.

Another example of mixing scriptlets and HTML is shown below -- here it is assumed that there is a boolean variable named "`hello`" available. If you set it to true, you will see one output, if you set it to false, you will see another output.

```
<%
    if ( hello ) {
        %>
        <P>Hello, world
        <%
    } else {
        %>
        <P>Goodbye, world
        <%
    }
%>
```

It is a little difficult to keep track of all open braces and scriptlet start and ends, but with a little practice and some good formatting discipline, you will acquire competence in doing it.

Exercise: Make the above examples work. Write a JSP to output all the values returned by `System.getProperties` with "
" embedded after each property name and value. Do not output the "
" using the "out" variable.

JSP Directives

We have been fully qualifying the `java.util.Date` in the examples in the previous sections. Perhaps you wondered why we don't just import java.util.*;

It is possible to use "`import`" statements in JSPs, but the syntax is a little different from normal Java. Try the following example:

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%
    System.out.println( "Evaluating date now" );
    Date date = new Date();
%>
Hello!  The time is now <%= date %>
</BODY>
</HTML>
```

The first line in the above example is called a "directive". A JSP "directive" starts with <%@ characters.

This one is a "page directive". The page directive can contain the list of all imported packages. To import more than one item, separate the package names by commas, e.g.

```
<%@ page import="java.util.*,java.text.*" %>
```

There are a number of JSP directives, besides the page directive. Besides the page directives, the other most useful directives are include and taglib. We will be covering taglib separately.

The include directive is used to physically include the contents of another file. The included file can be HTML or JSP or anything else -- the result is as if the original JSP file actually contained the included text. To see this directive in action, create a new JSP

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<%@ include file="hello.jsp" %>
</BODY>
</HTML>
```

View this JSP in your browser, and you will see your original `hello.jsp` get included in the new JSP.

Exercise: Modify all your earlier exercises to import the java.util packages.

JSP Declarations

The JSP you write turns into a class definition. All the scriptlets you write are placed inside a single method of this class.

You can also add variable and method declarations to this class. You can then use these variables and methods from your scriptlets and expressions.

To add a declaration, you must use the **<%!** and **%>** sequences to enclose your declarations, as shown below.

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%!
    Date theDate = new Date();
    Date getDate()
    {
        System.out.println( "In getDate() method" );
        return theDate;
    }
%>
Hello!  The time is now <%= getDate() %>
</BODY>
</HTML>
```

The example has been created a little contrived, to show variable and method declarations.

Here we are declaring a Date variable `theDate`, and the method `getDate`. Both of these are available now in our scriptlets and expressions.

But this example no longer works! The date will be the same, no matter how often you reload the page. This is because these are declarations, and will only be evaluated once

when the page is loaded! (Just as if you were creating a class and had variable initialization declared in it.)

Exercise: Modify the above example to add another function `computeDate` which re-initializes `theDate`. Add a scriptlet that calls `computeDate` each time.

Note: Now that you know how to do this -- it is in general not a good idea to use variables as shown here. The JSP usually will run as multiple *threads* of one single instance. Different threads would interfere with variable access, because it will be the same variable for all of them. If you do have to use variables in JSP, you should use *synchronized* access, but that hurts the performance. In general, any data you need should go either in the *session* object or the *request* object (these are introduced a little later) if passing data between different JSP pages. Variables you declare inside *scriptlets* are fine, e.g. <% int i = 45; %> because these are declared inside the local scope and are not shared.

JSP Tags

Another important syntax element of JSP are tags. JSP tags do not use **<%**, but just the **<** character. A JSP tag is somewhat like an HTML tag. JSP tags can have a "start tag", a "tag body" and an "end tag". The start and end tag both use the tag name, enclosed in < and > characters. The end starts with a / character after the < character. The tag names have an embedded colon character : in them, the part before the colon describes the type of the tag. For instance:

```
<some:tag>
body
</some:tag>
```

If the tag does not require a body, the start and end can be conveniently merged together, as

```
<some:tag/>
```

Here by closing the start tag with a /> instead of > character, we are ending the tag immediately, and without a body. (This syntax convention is the the same as XML.)

Tags can be of two types: loaded from an external tag library, or predefined tags. Predefined tags start with **`jsp:`** characters. For instance, `jsp:include` is a predefined tag that is used to include other pages.

We have already seen the `include` directive. jsp:include is similar. But instead of loading the text of the included file in the original file, it actually calls the included target at run-time (the way a browser would call the included target. In practice, this is actually a simulated request rather than a full round-trip between the browser and the server). Following is an example of `jsp:include` usage

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<jsp:include page="hello.jsp"/>
</BODY>
</HTML>
```

Try it and see what you get. Now change the "`jsp:include`" to "`jsp:forward`" and see what is the difference. These two predefined tags are frequently very useful.

Exercise: Write a JSP to do either a `forward` or an `include`, depending upon a boolean variable (hint: The concepts of mixing HTML and scriptlets work with JSP tags also!)

JSP Sessions

On a typical web site, a visitor might visit several pages and perform several interactions.

If you are programming the site, it is very helpful to be able to associate some data with each visitor. For this purpose, "`session`"s can be used in JSP.

A session is an object associated with a visitor. Data can be put in the session and retrieved from it, much like a Hashtable. A different set of data is kept for each visitor to the site.

Here is a set of pages that put a user's name in the session, and display it elsewhere. Try out installing and using these.

First we have a form, let us call it GetName.html

```
<HTML>
<BODY>
<FORM METHOD=POST ACTION="SaveName.jsp">
What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20>
<P><INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

The target of the form is "SaveName.jsp", which saves the user's name in the session. Note the variable "`session`". This is another variable that is normally made available in JSPs, just like `out` and `request` variables. (In the @page directive, you can indicate that you do not need sessions, in which case the "session" variable will not be made available.)

```
<%
   String name = request.getParameter( "username" );
   session.setAttribute( "theName", name );
%>
<HTML>
<BODY>
<A HREF="NextPage.jsp">Continue</A>
</BODY>
</HTML>
```

The SaveName.jsp saves the user's name in the session, and puts a link to another page, NextPage.jsp.

NextPage.jsp shows how to retrieve the saved name.

```
<HTML>
<BODY>
Hello, <%= session.getAttribute( "theName" ) %>
</BODY>
</HTML>
```

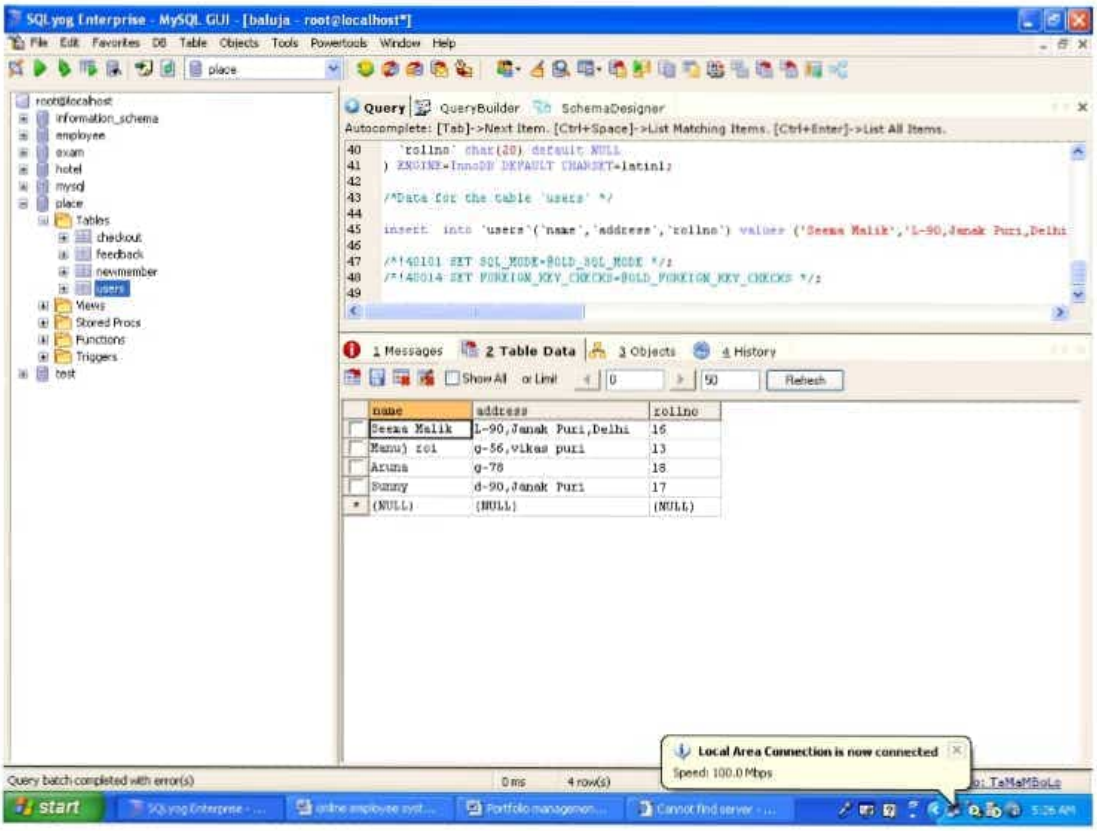
If you bring up two different browsers (not different windows of the same browser), or run two browsers from two different machines, you can put one name in one browser and another name in another browser, and both names will be kept track of.

The session is kept around until a timeout period. Then it is assumed the user is no longer visiting the site, and the session is discarded.

MY SQL:

Introduction

My SQL is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA). MY SQL is probably the most accessible and the most documented enterprise database environment right now. This also means that you can learn it a little quicker than most other database environments on the market



To start, you must have a computer that runs an appropriate operating system like Microsoft Windows >= XP Home Edition: that includes Windows XP Home Edition, Windows XP

AIMS & OBJECTIVES OF THE STUDY

Objective:

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as "ADMINISTRATION SUPPORT SYSTEM FOR MEDICAL INSTITUTIONS".

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

AIM:

The aim of the study to fully related with Hospital Management system.

- The Software is for the automation of Hospital Management System.
- It maintains two levels of users:-

_ Administrator Level

_ User Level

- The Software includes:-

_ Maintaining Patient details.

_ Providing Prescription, Precautions and Diet advice.

_ Providing and maintaining all kinds of tests for a patient.

_ Billing and Report generation.

RESEARCH METHODOLOGY

The project 'Hospital Management System' is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses JAVA as the front-end software which is an Object Oriented Programming and has connectivity with MY SQL. It is a web based application in which number of clients can also access with a server.

HARDWARE

Processor	:	Pentium 2.4 GHz or above
Memory	:	256 MB RAM or above
Cache Memory	:	128 KB or above
Hard Disk	:	3 GB or above [at least 3 MB free space required]
Pen Drive	:	**5 GB**
Printer	:	Laser Printer

SOFTWARE

Operating System	:	Windows XP (Professional).
Font-End Tool	:	JSP, Servlets, Java Script
Back-End	:	My Sql

FRONT END

We have implemented **JavaScript** for all the Client side validations. Client side JavaScript is designed to reside inside HTML document & ensure they run properly. It is object based, event driven, platform independent. These are important parts of any Web application to implement Client side Validations and the invalid data is not submitted. The form is not submitted until user fills in correct data. It is extremely useful to restrict mistakes by user.

BACK END

We have used My Sql. My Sql provides efficient/effective solution for major database tech.

- Large database and space management.
- Many concurrent database users.
- High transaction processing requirement
- High Availability
- Industry accepted standards
- Manageable security
- Portability

SYSTEM ANALYSIS

PRINCIPLES OF SYSTEM ANALYSIS:

PRINCIPLES:

- Understand the problem before you begin to create the analysis model.
- Develop prototypes that enable a user to understand how human machine interaction will occur.
- Record the origin of and the reason for every requirement.
- Use multiple views of requirements like building data, function and behavioral models.
- Work to eliminate ambiguity

System Analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system. One of the main causes of project

failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- What type of information is required?
- What are the constraints on the investigation?
- What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system; the total system was designed and can be described as under:

The three major parts of the system are:

- **Providing Information:**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide access to records of various Job seekers & users of matrimonial such as resume & profile of boys and girls those who want to search a life partner with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as a information provider for company & life partner seekers.

Preliminary Investigation

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organisation initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

1) Request clarification
2) Feasibility study
3) Request approval

Request clarification: Many requests from employees and users in the organisations are not clearly defined, Therefore it becomes necessary that project request must be eximined and clarified properly before considering systems investigation.

FEASIBILITY STUDY:

The feasibility study proposes one or more conceptual solution to the problem set of the project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or not.

1. Evaluation of feasibility of such solutions. Such evaluation often indicates shortcomings in the initial goals. This step is repeated as the goals are adjusted and the alternative solutions are evaluated.

Feasibility analysis usually considers a number of project alternatives, one that is chosen as the most satisfactory solution. These alternatives also need to be evaluated in a broad way without committing too many resources. Various steps involved in feasibility analysis are:

2. To propose a set of solution that can realize the project goal. These solutions are usually descriptions of what the new system should look like.

Four primary areas of interest in feasibility study are:

Economic Feasibility: An evaluation of development cost weighed against the ultimate income of benefit derived from the development system of product. In economic feasibility, cost benefit analysis is done in which expected cost and benefits are evaluated.

information. This method is however less effective for learning about people's perceptions, feelings and motivations.

ANALYST'S INTERVIEW WITH HOSPITAL ADMINISTRATOR:

Analyst: Hi, I have come to talk to you regarding the functioning of your hospital project.

Administrator: hello, do come in. I was expecting you.

Analyst: I'll come straight to the point. Don't hesitate, you can be as much open you want. There are no restrictions.

Administrator: I'll give you my whole contribution.

Analyst: Tell me are you excited about the idea of having an automated system for your hospital?

Administrator: Yes, I do. Very much. After all it's gonna reduce our loads of work.

Analyst: Will you elaborate on it?

Administrator: Major problem is managing the rooms and admission and discharge of patients. There are so many of rooms and the numbers of patients are admitted in the room. At the time of discharging of a patient, it becomes more difficult to handle the rooms condition, and re-admitting the patients in that room.

Analyst: What do you think be ideal solution to this?

Administrator: All the information of rooms, patients and discharged patients should be put into computer. It'll be easy for us to check how

many rooms are not available and which patient has to be allotted what room.

Analyst: Could you explain how?

Administrator: Look whenever a new patient is admitted he/she is allotted a bed number and the bed is reserved for the patient till the patient gets discharged. And when the patient is discharged, the bed allotted to him/her is freed and now the bed should again automatically ready for new patient to be admitted.

Analyst: Do you have different patient categories?

Administrator: No, we don't have any categorization for patients. All are treated at par.

Analyst: How do you categorize your patients?

Administrator: By Bed number.

Analyst: Do you have any other expectations or suggestion for the new system?

Administrator: It should be able to produce reports faster.

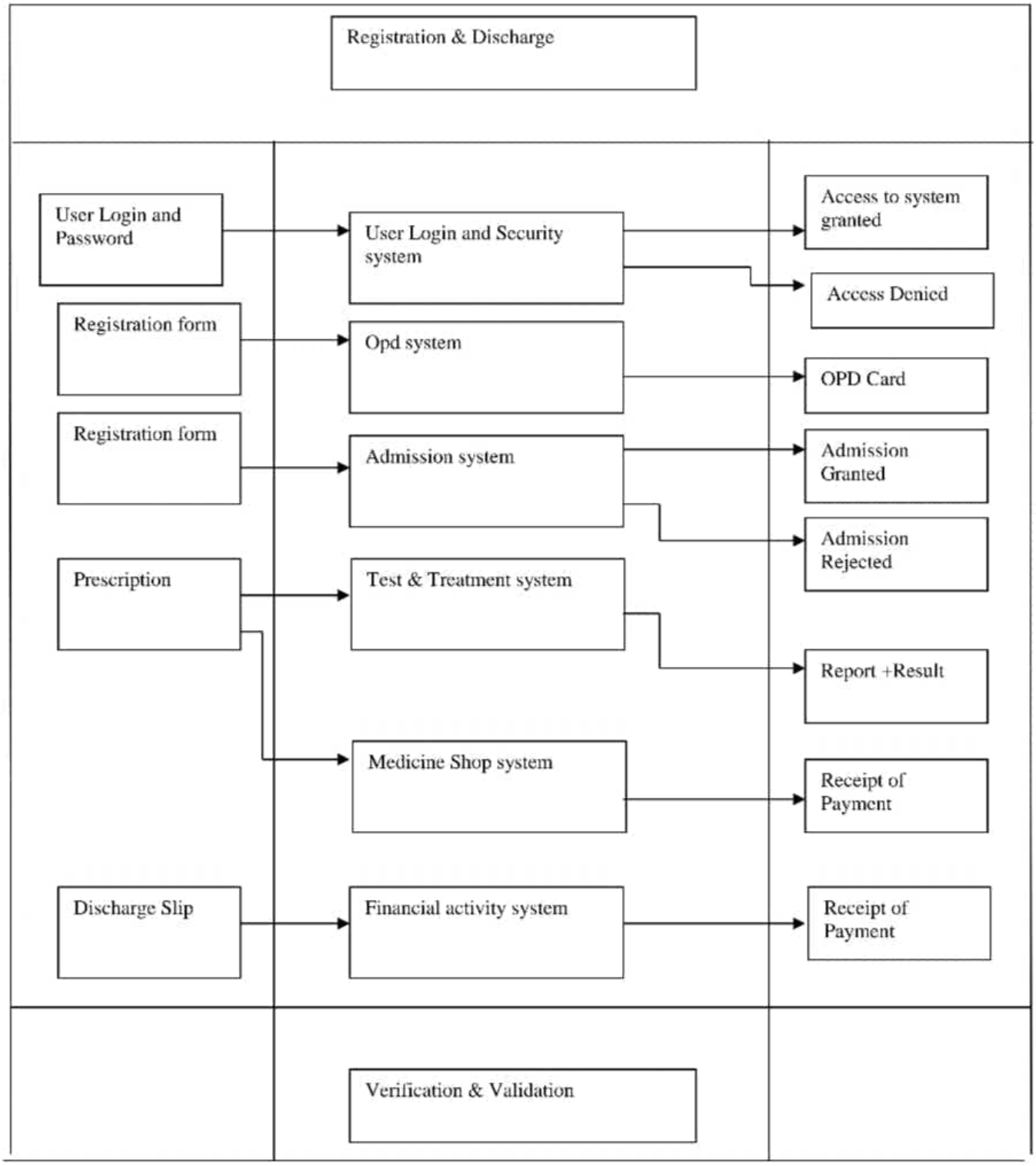
Analyst: Reports? I completely forgot about them. What reports you people produce presently?

Administrator: Well first is for room status, another for patient's list being admitted and discharged patients and reports for doctors.

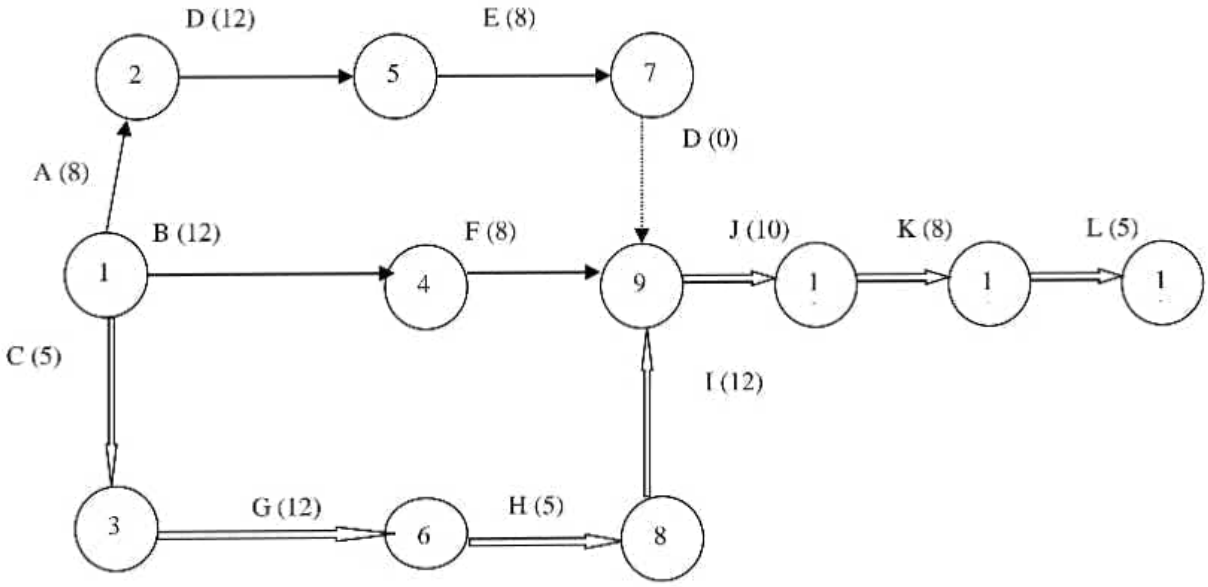
Analyst: Do you have some format for them?

Administrator: Yes we do have and we want that the same format be used by the new system.

ACD (Architectural Context Diagram

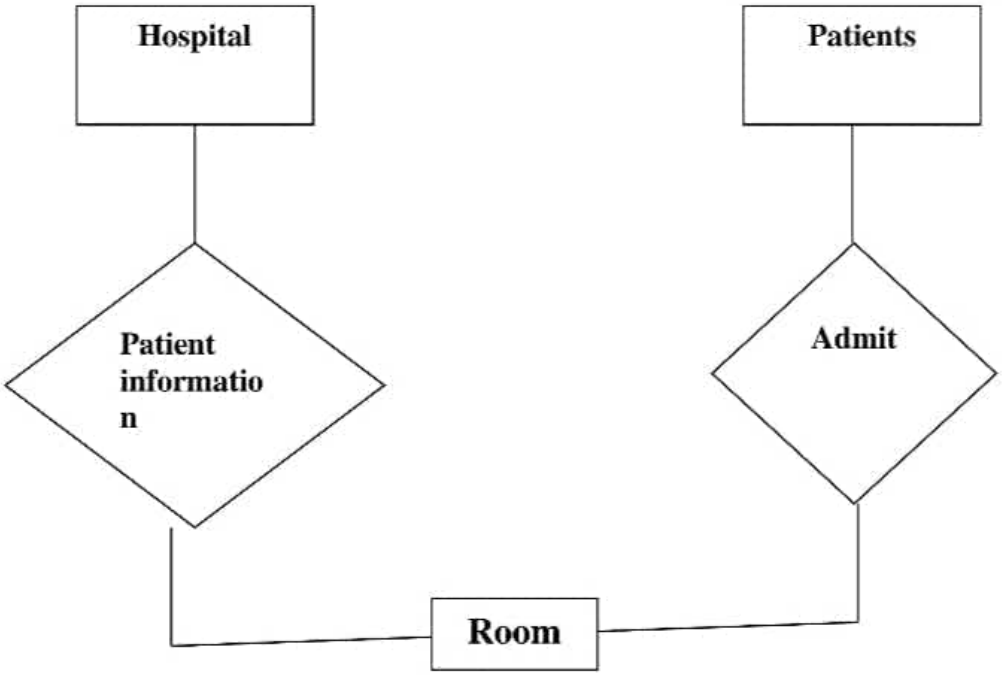


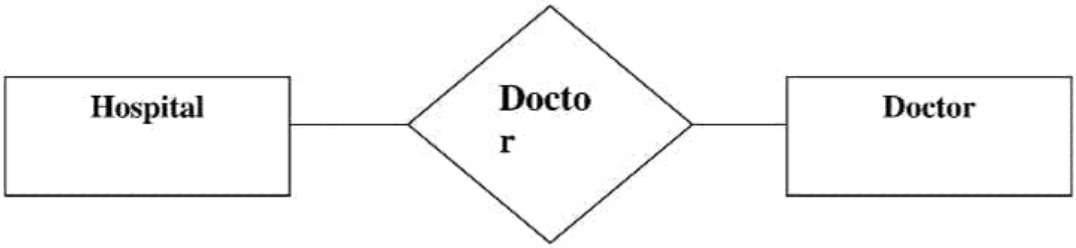
PERT CHART

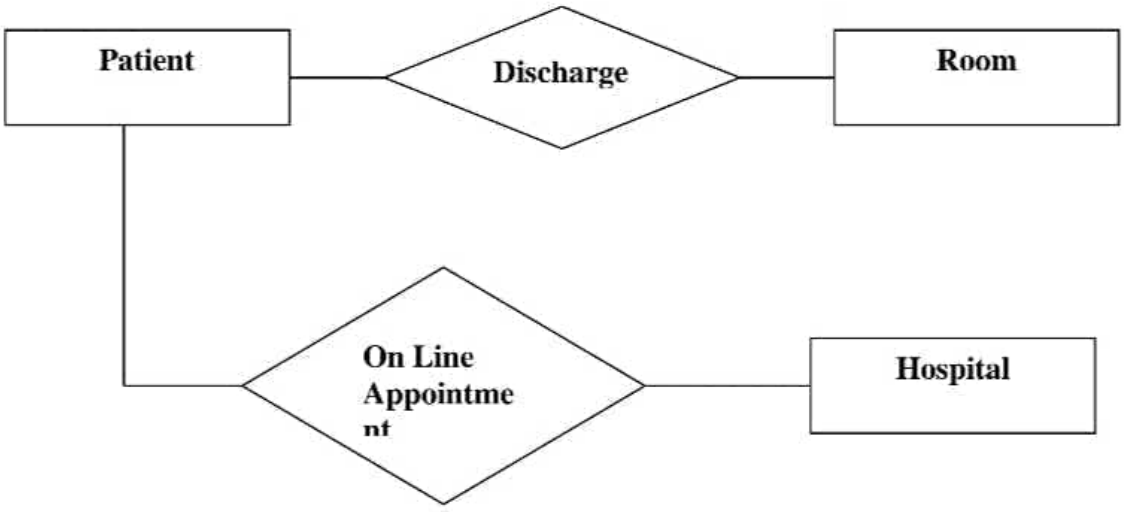


	Critical Path
	Activity
	Dummy Activity
	Node

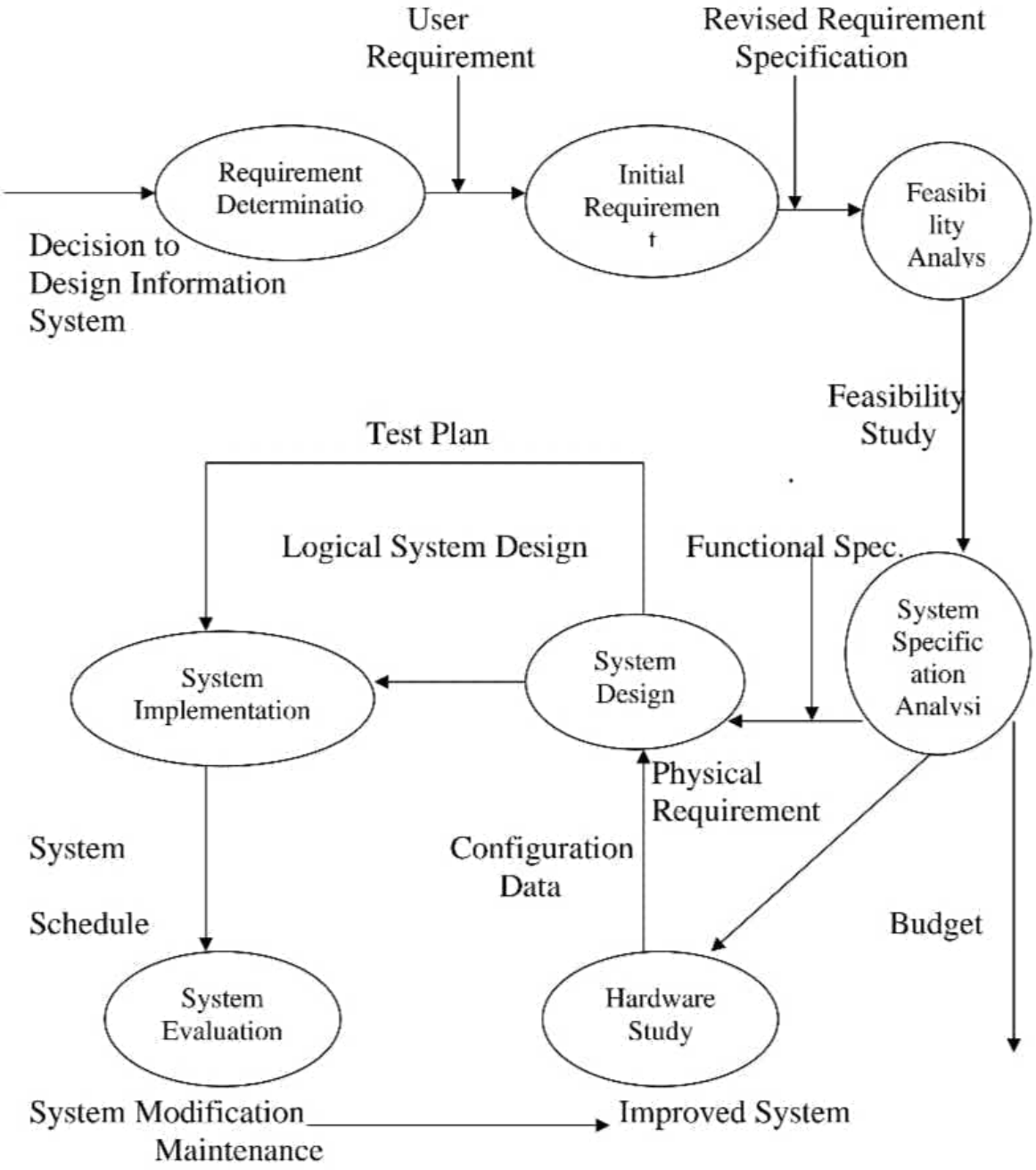
E-R- DIAGRAM


Hospital
Patients
Patient
informatio
n
Admit
Room


Hospital
Docto
r
Doctor

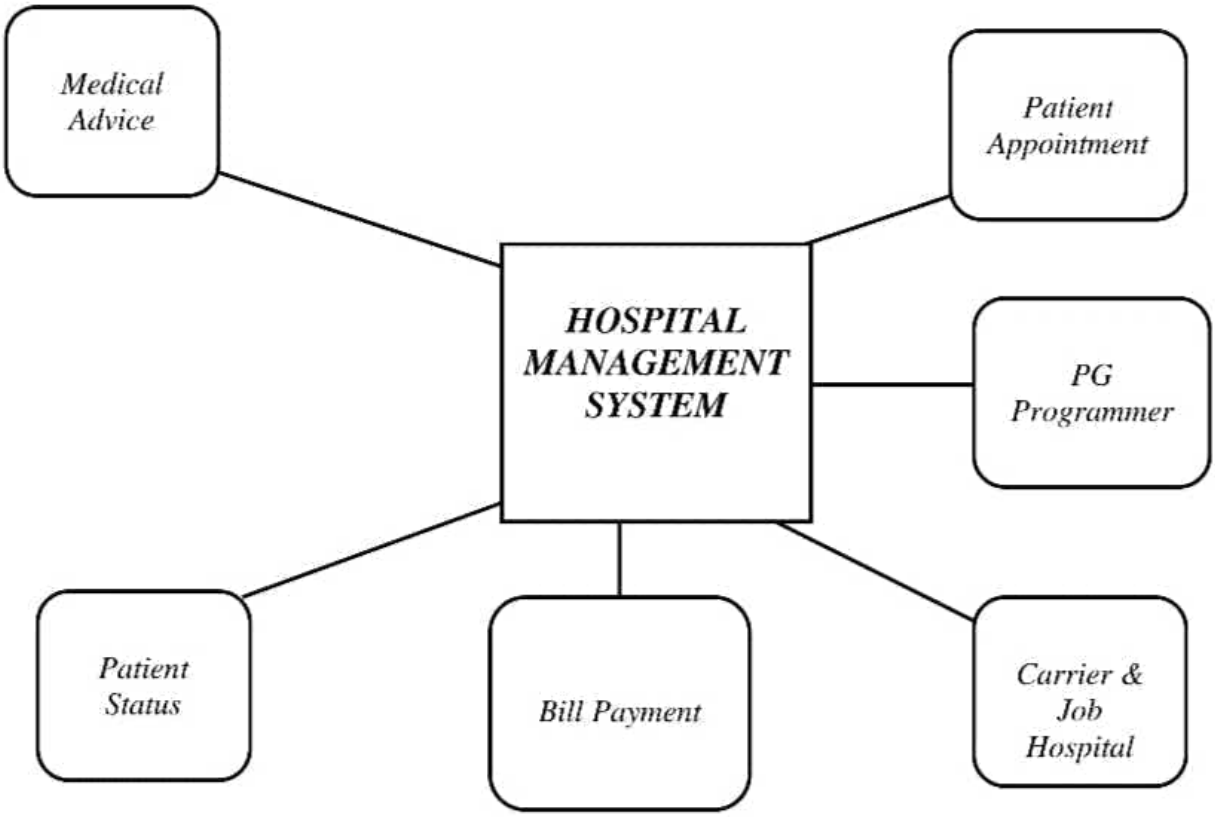

Patient
Discharge
Room
On Line
Appointme
nt
Hospital

SYSTEM DEVELOPMENT LIFE CYCLE

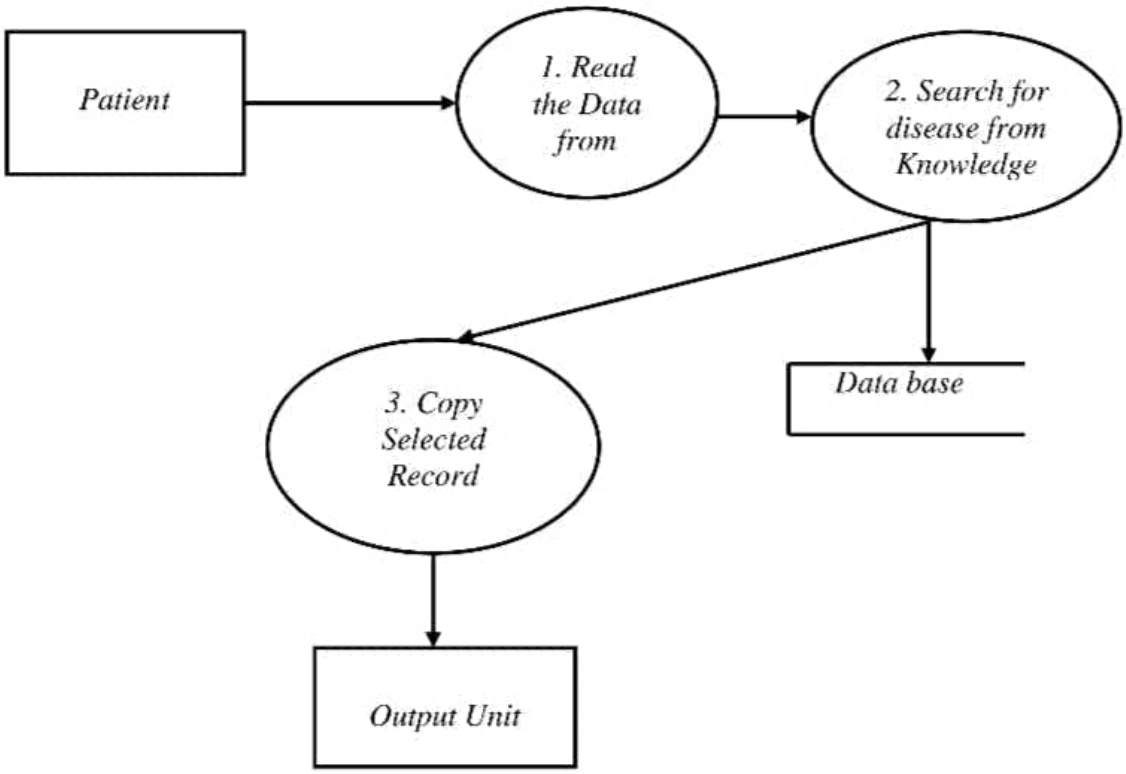


HOSPITALMANAGEMENT SYSTEM
DATA FLOW DIAGRAM

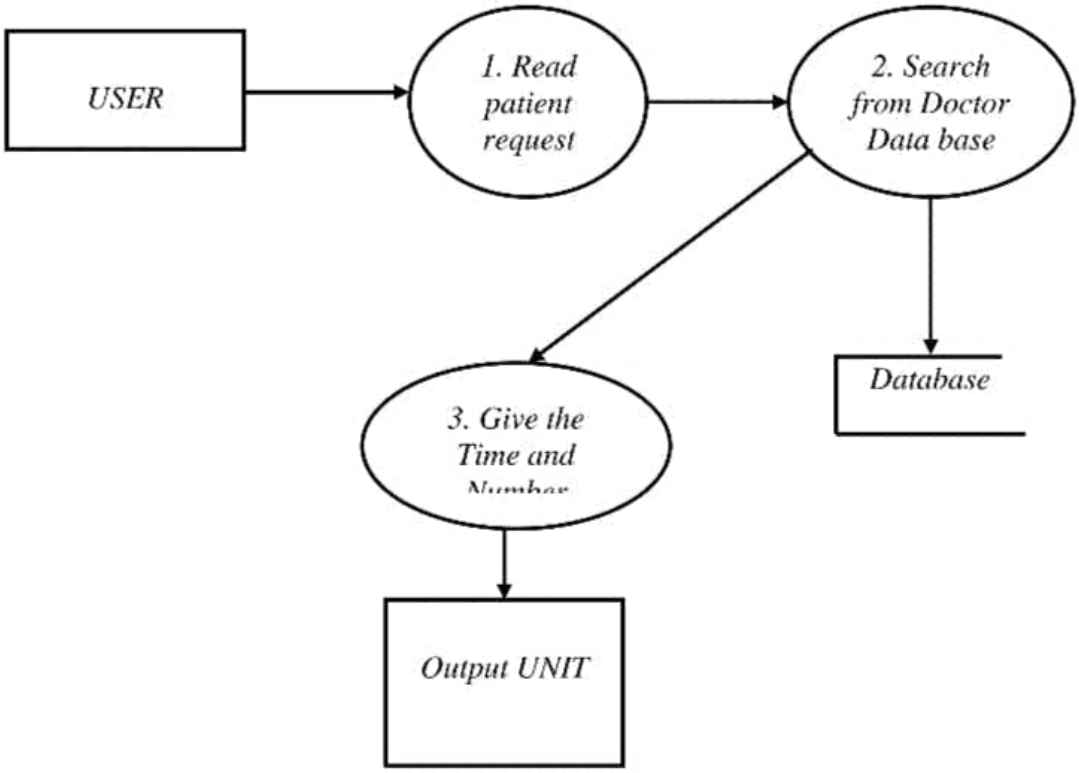
Context Level DFD



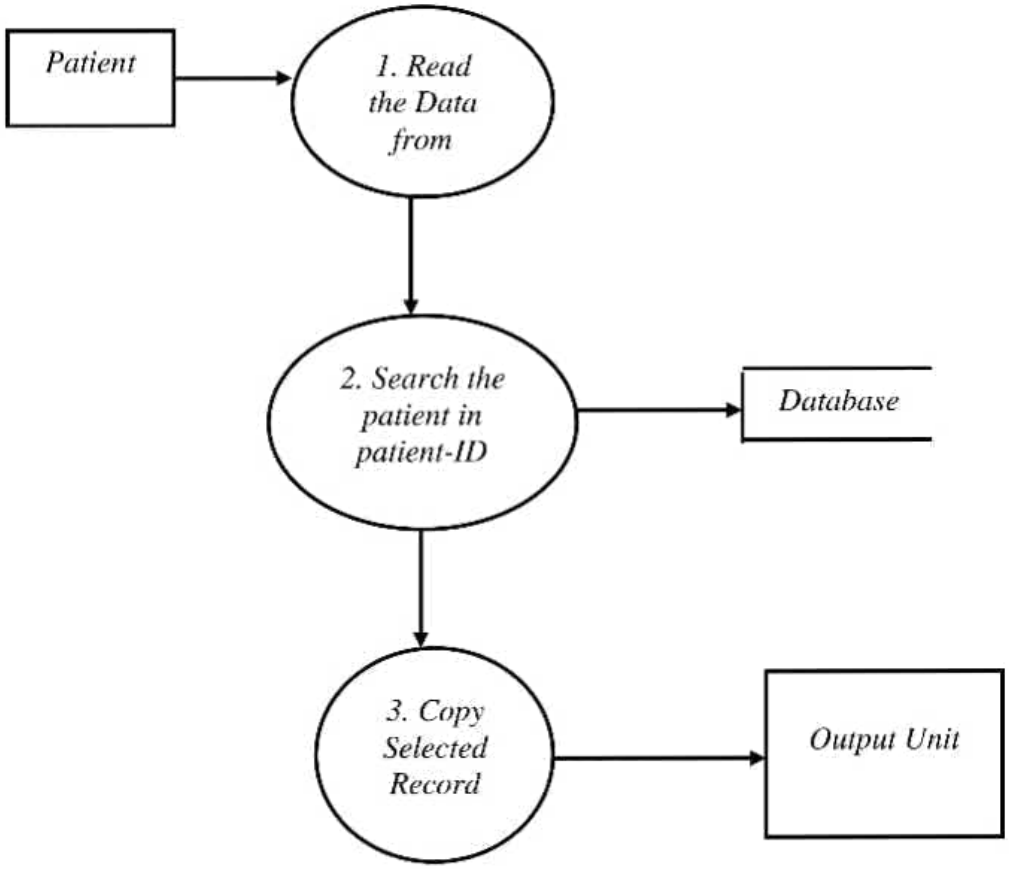
DFD for Medical Advice



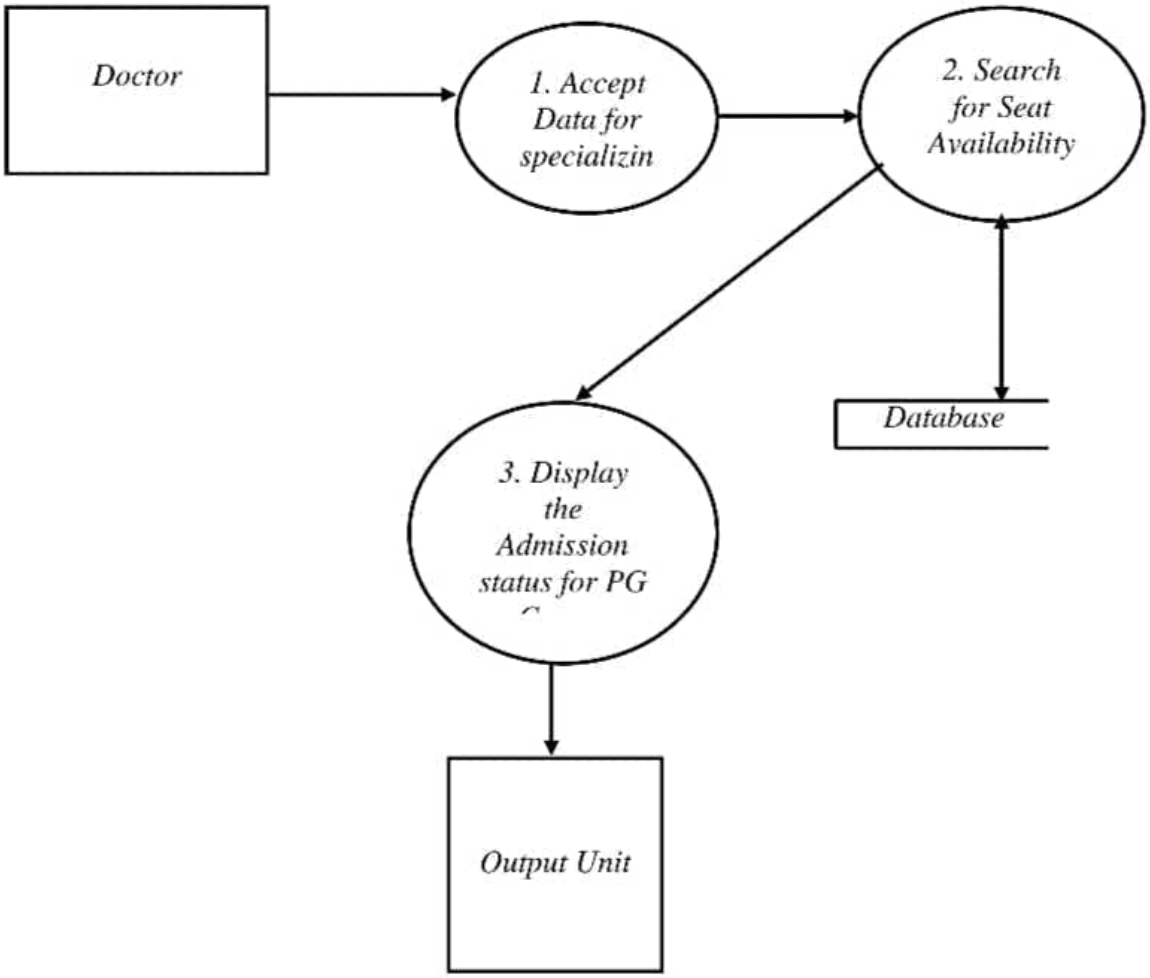
DFD for patient Appointment



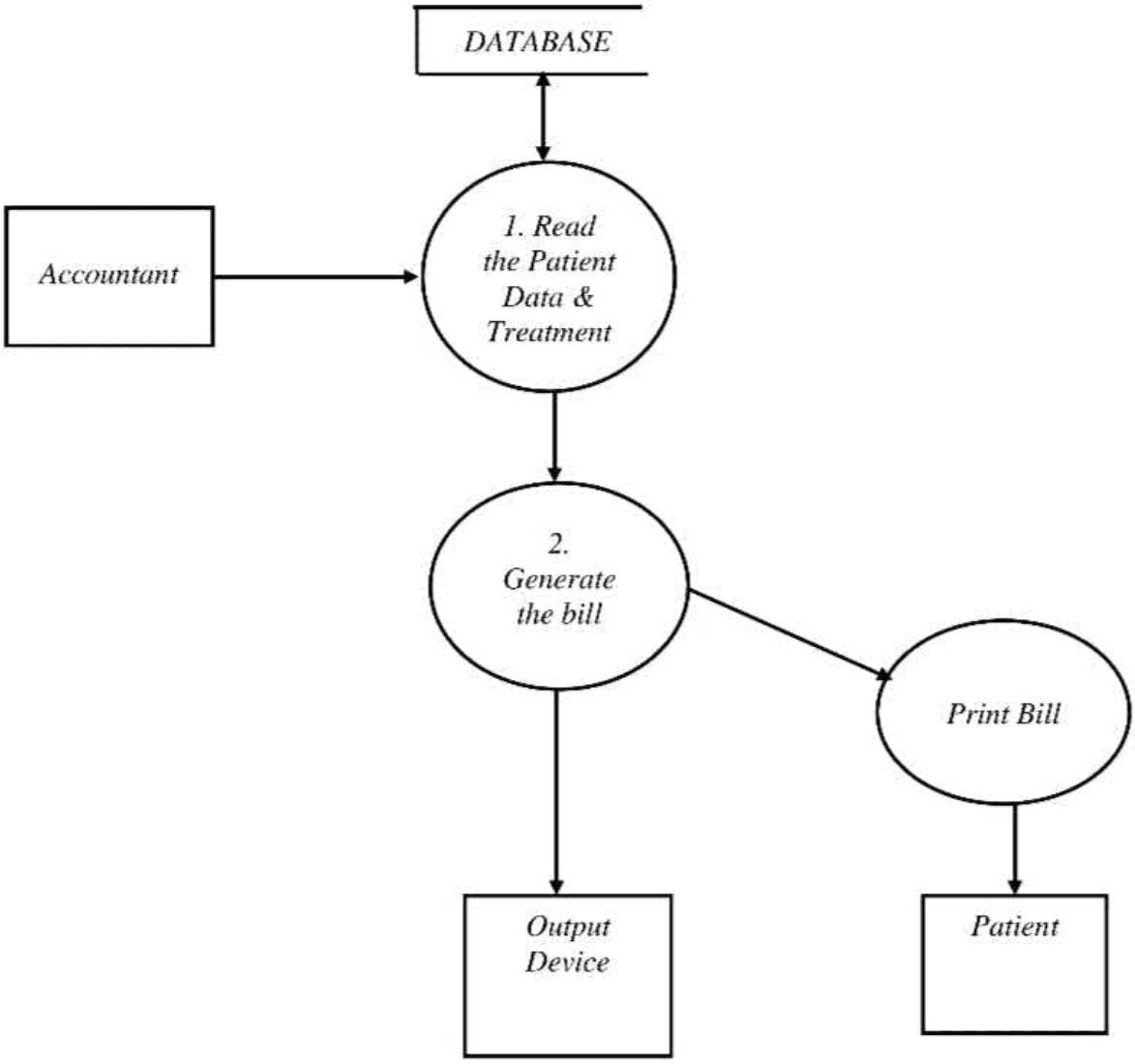
DFD for Patient Search



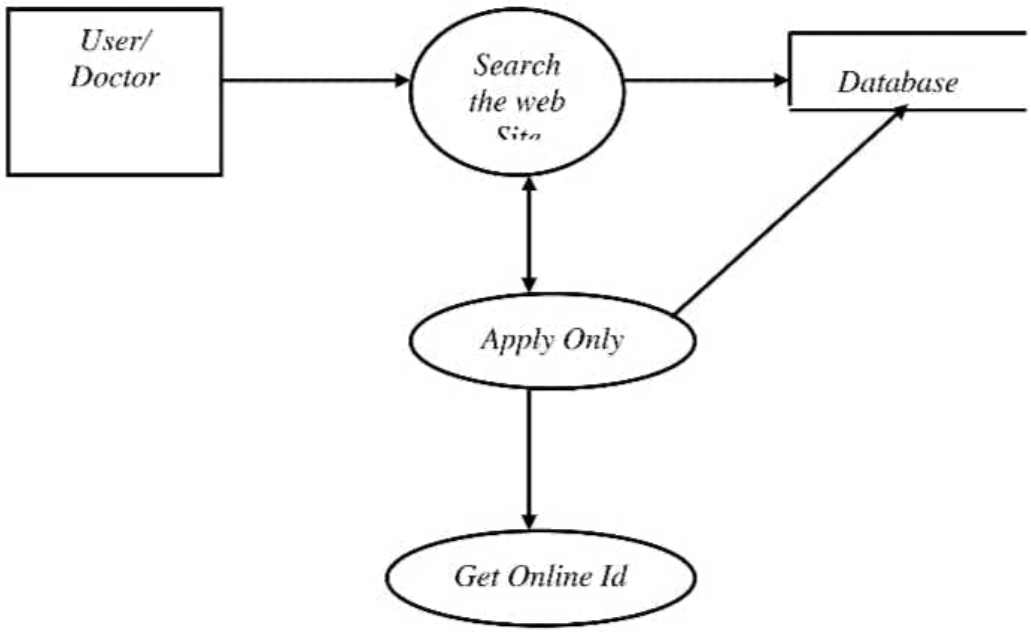
DFD for PG Course



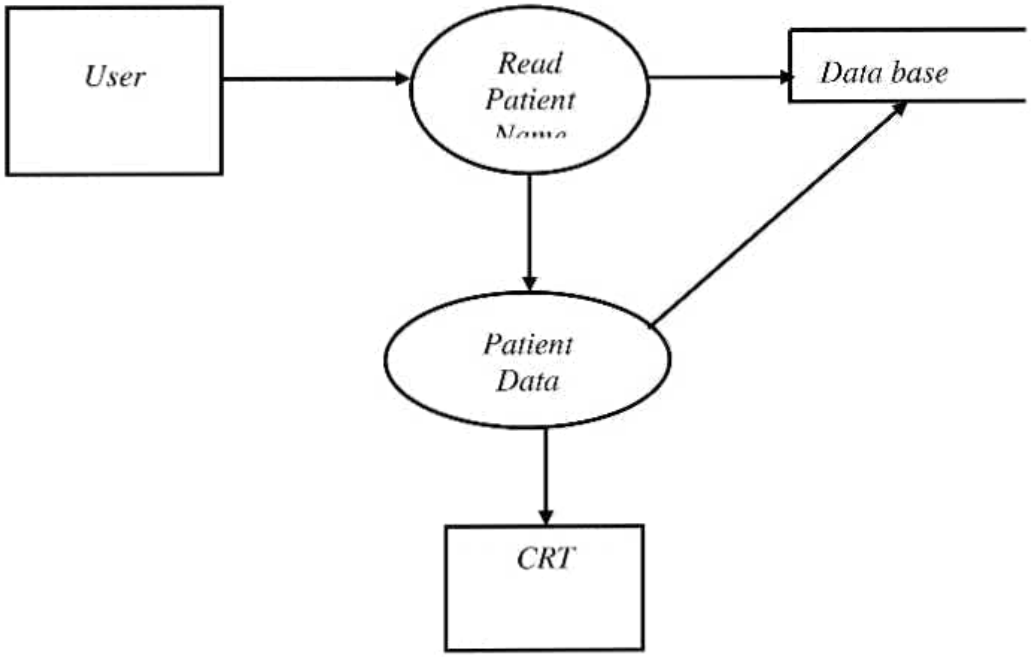
DFD For Bill Payment



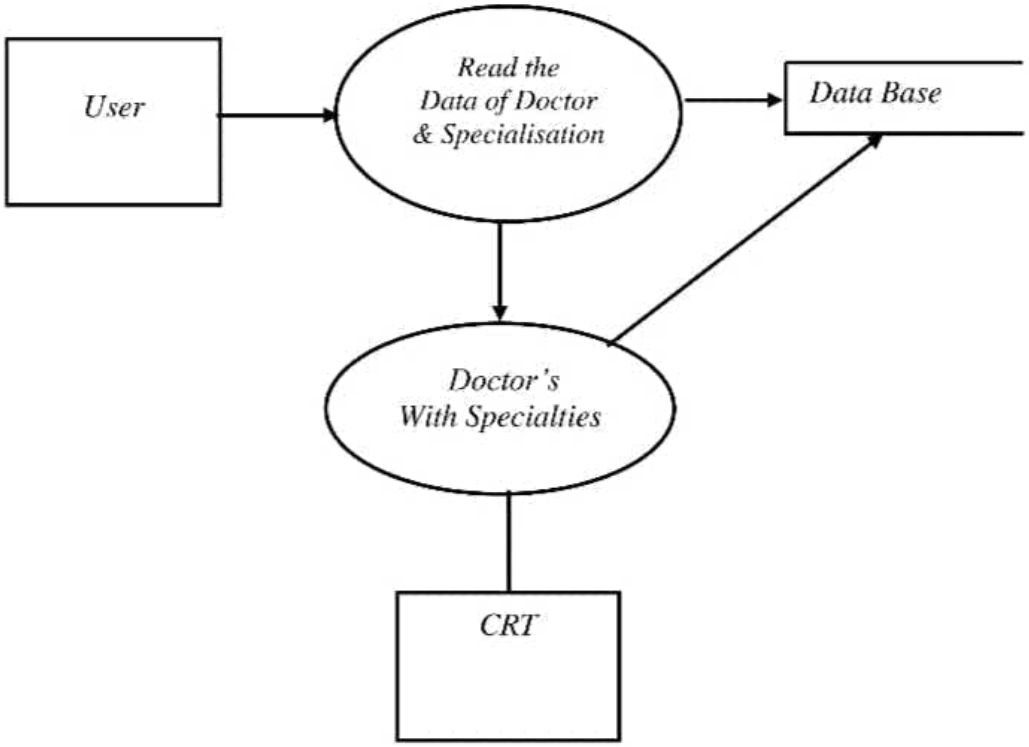
DFD For Job Opportunity In hospital



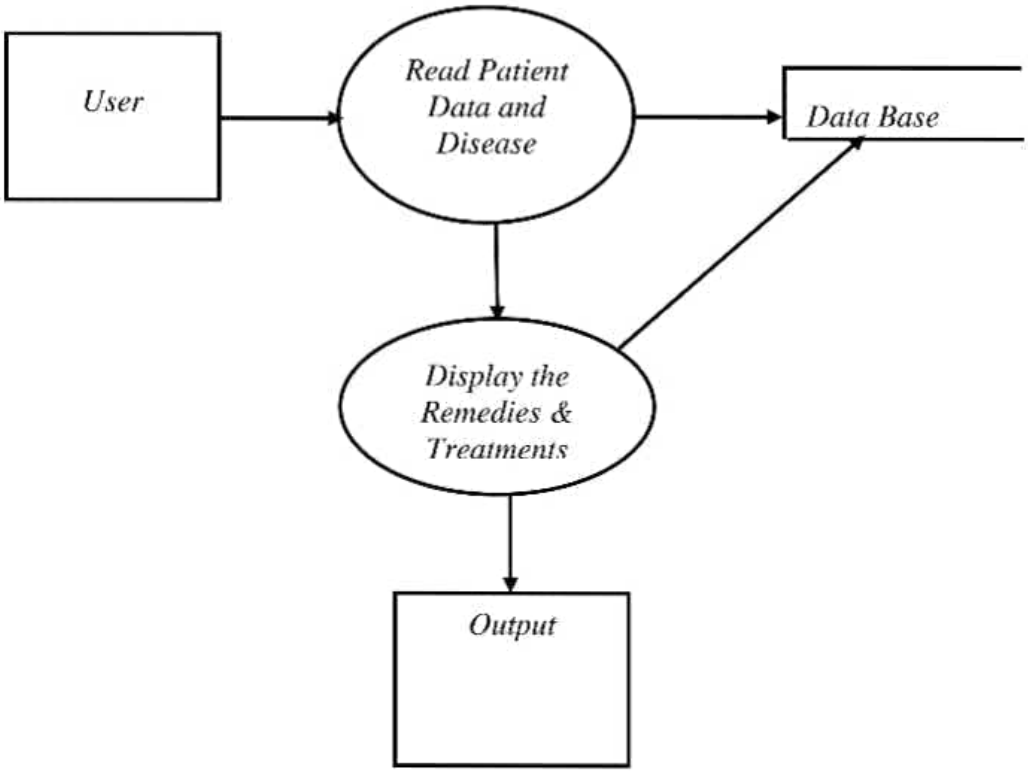
DFD For Online Searching For Patient



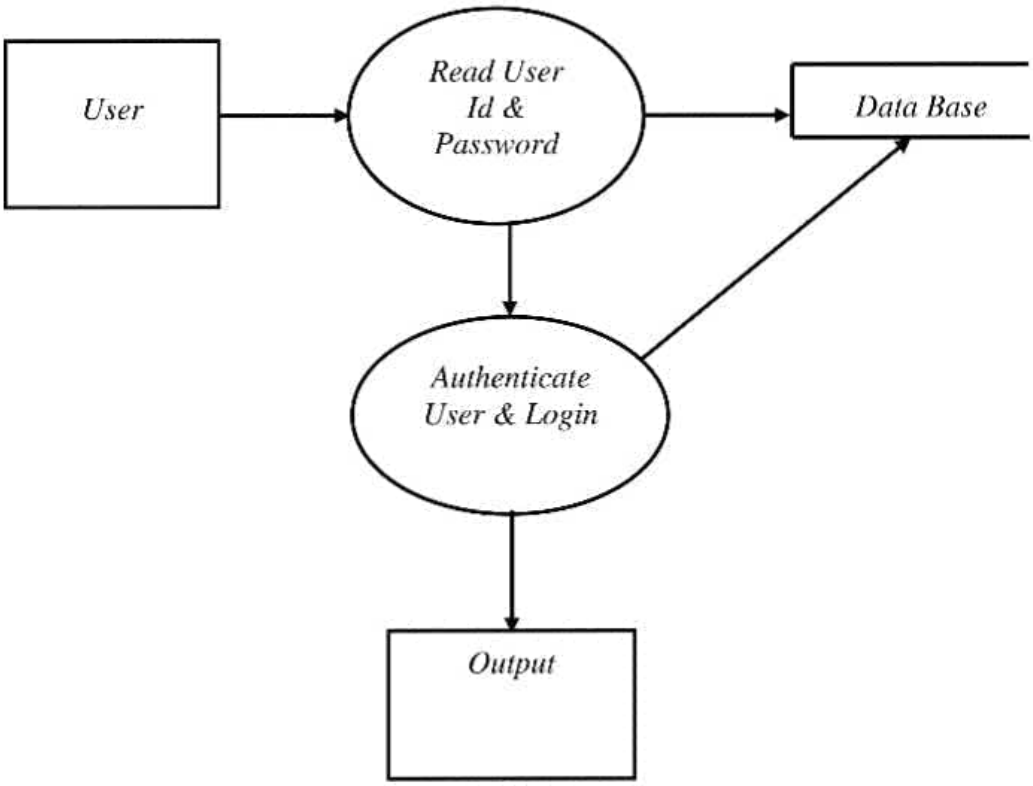
DFD For Searching a Doctors



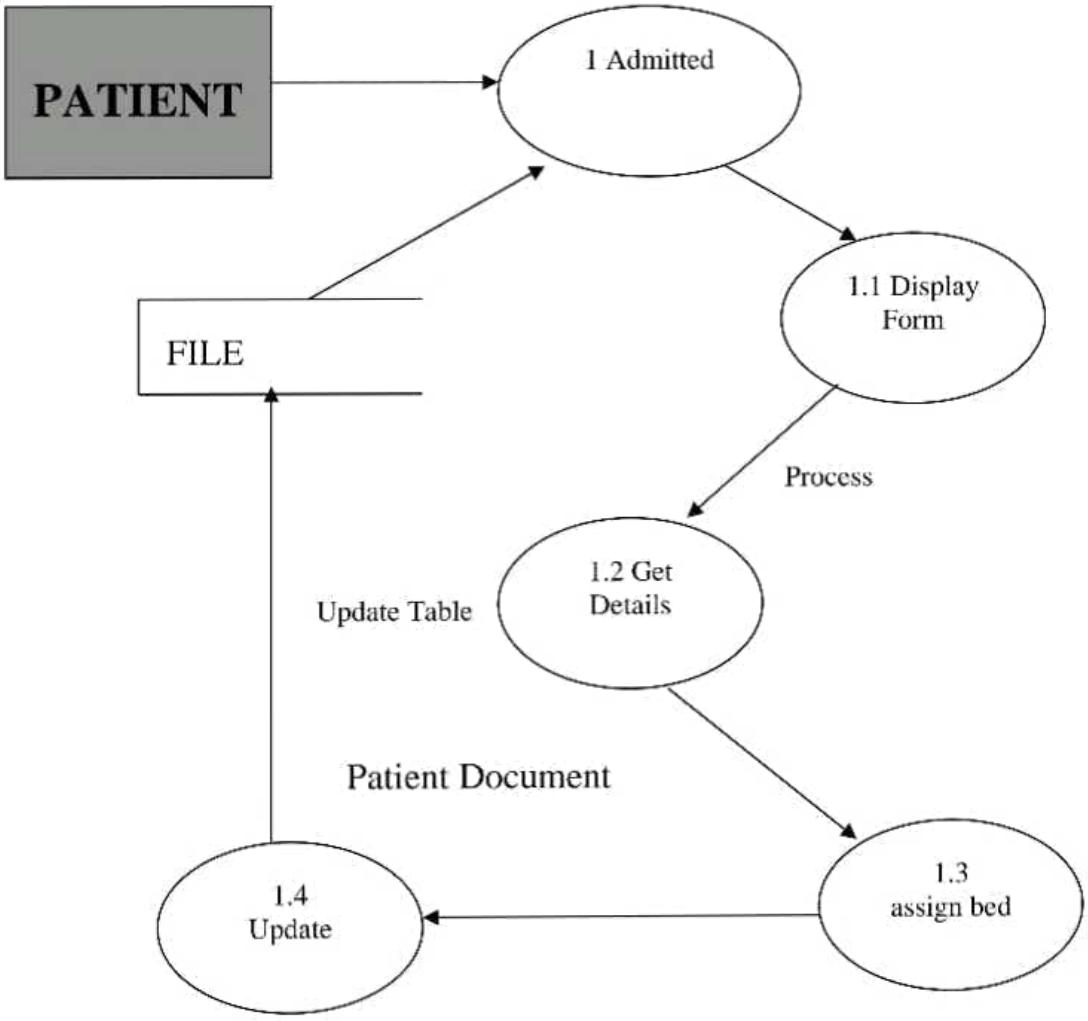
DFD Online Medical Advice



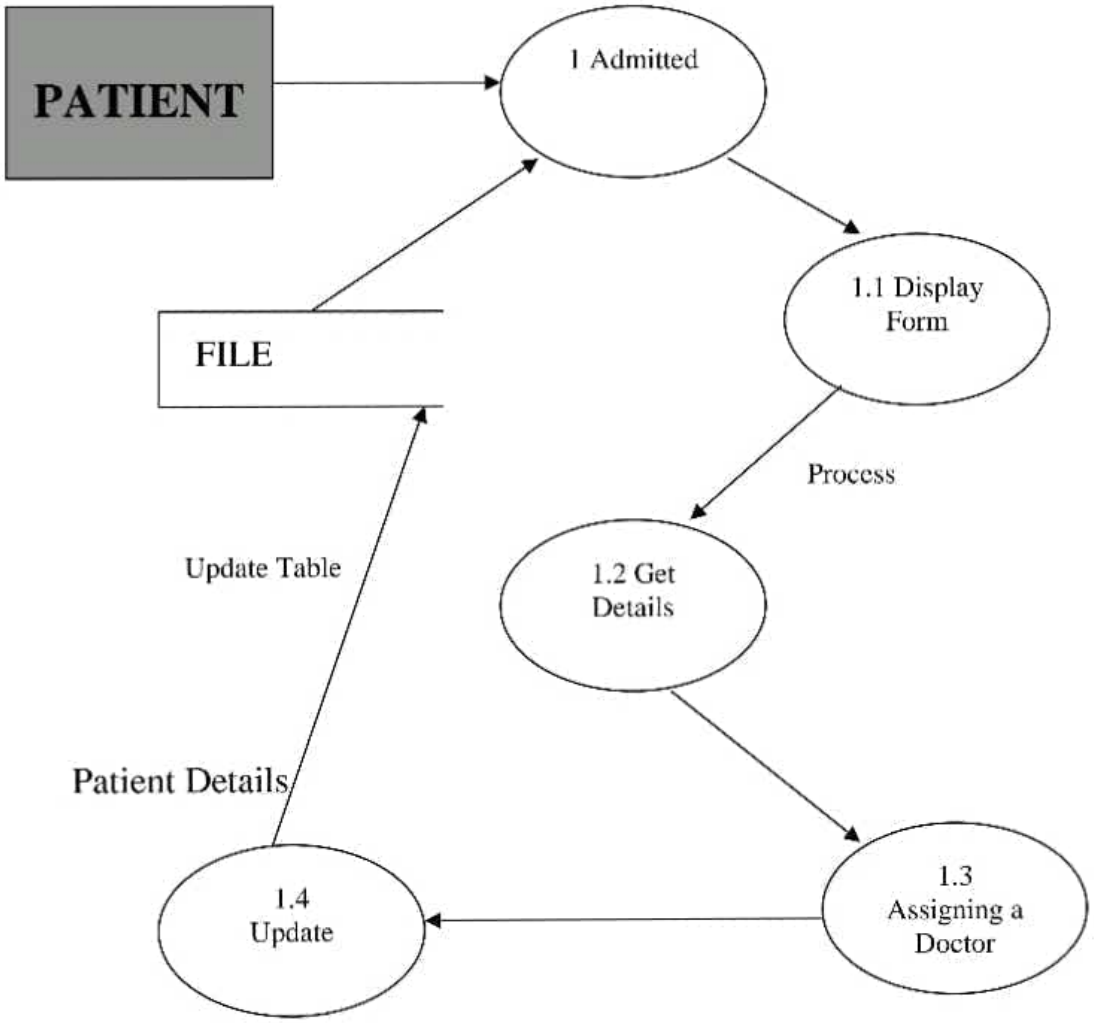
DFD For Login Of User



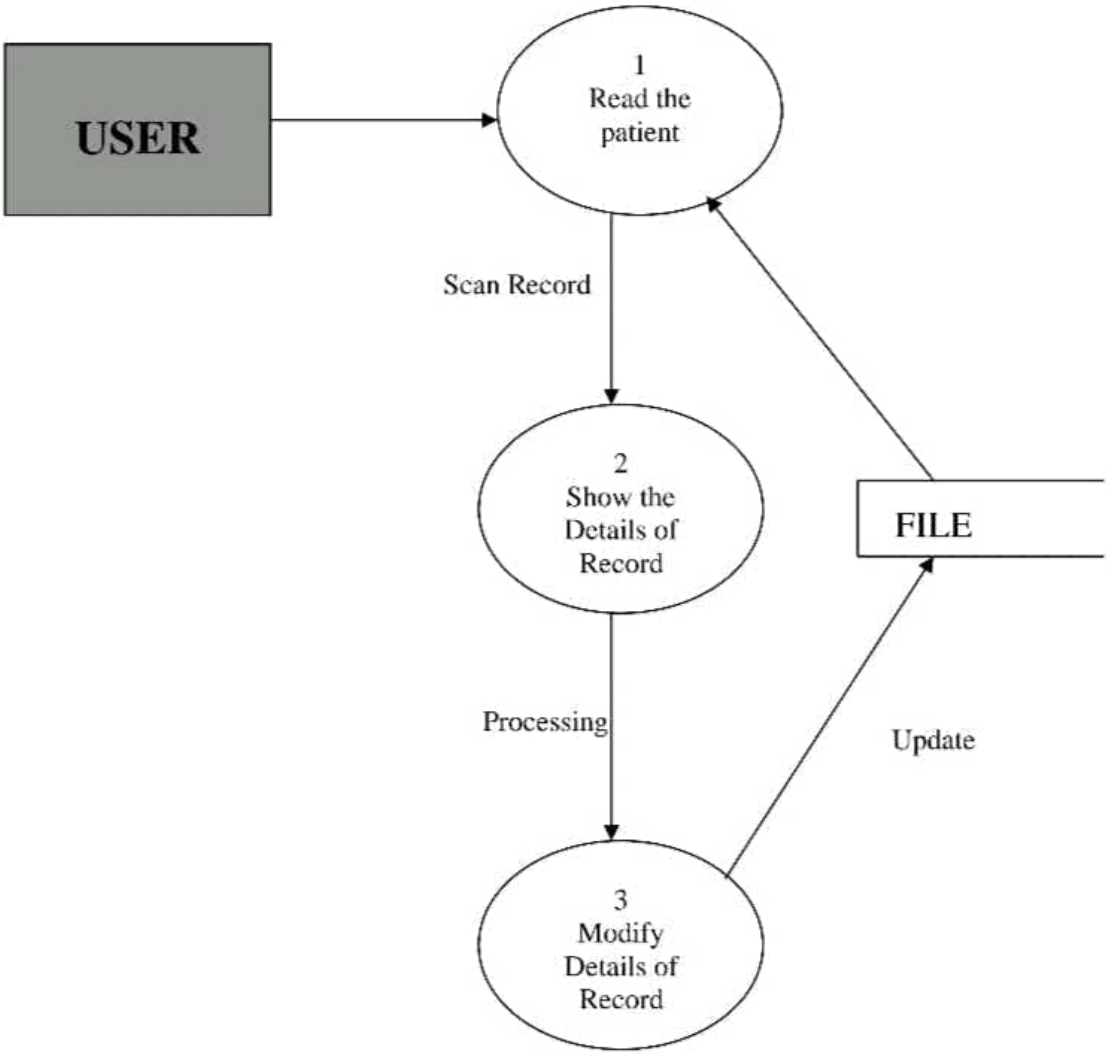
Bed Details


PATIENT
1 Admitted
1.1 Display Form
FILE
Process
1.2 Get Details
Update Table
Patient Document
1.3 assign bed
1.4 Update

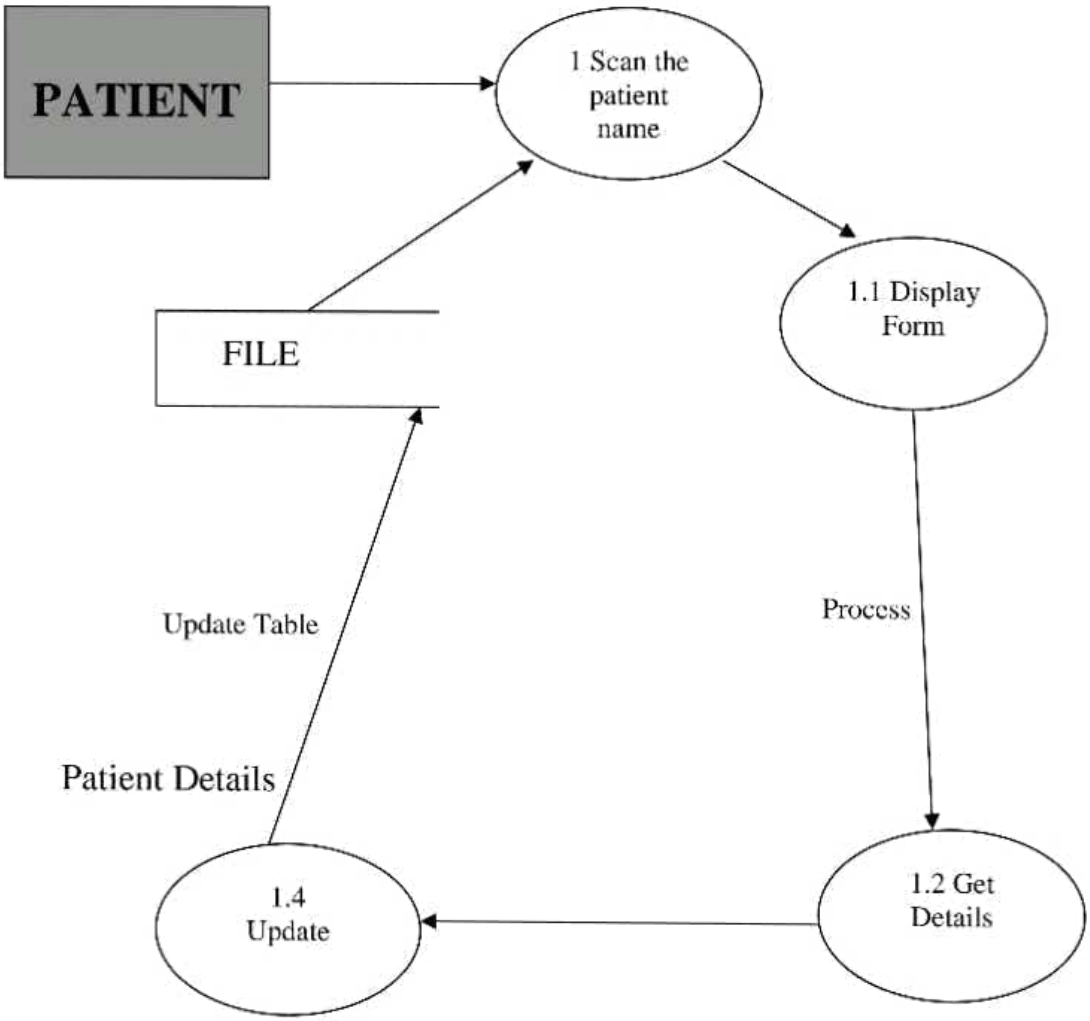
DATA FLOW DIAGRAM
ADMISSION OF A NEW PATIENT



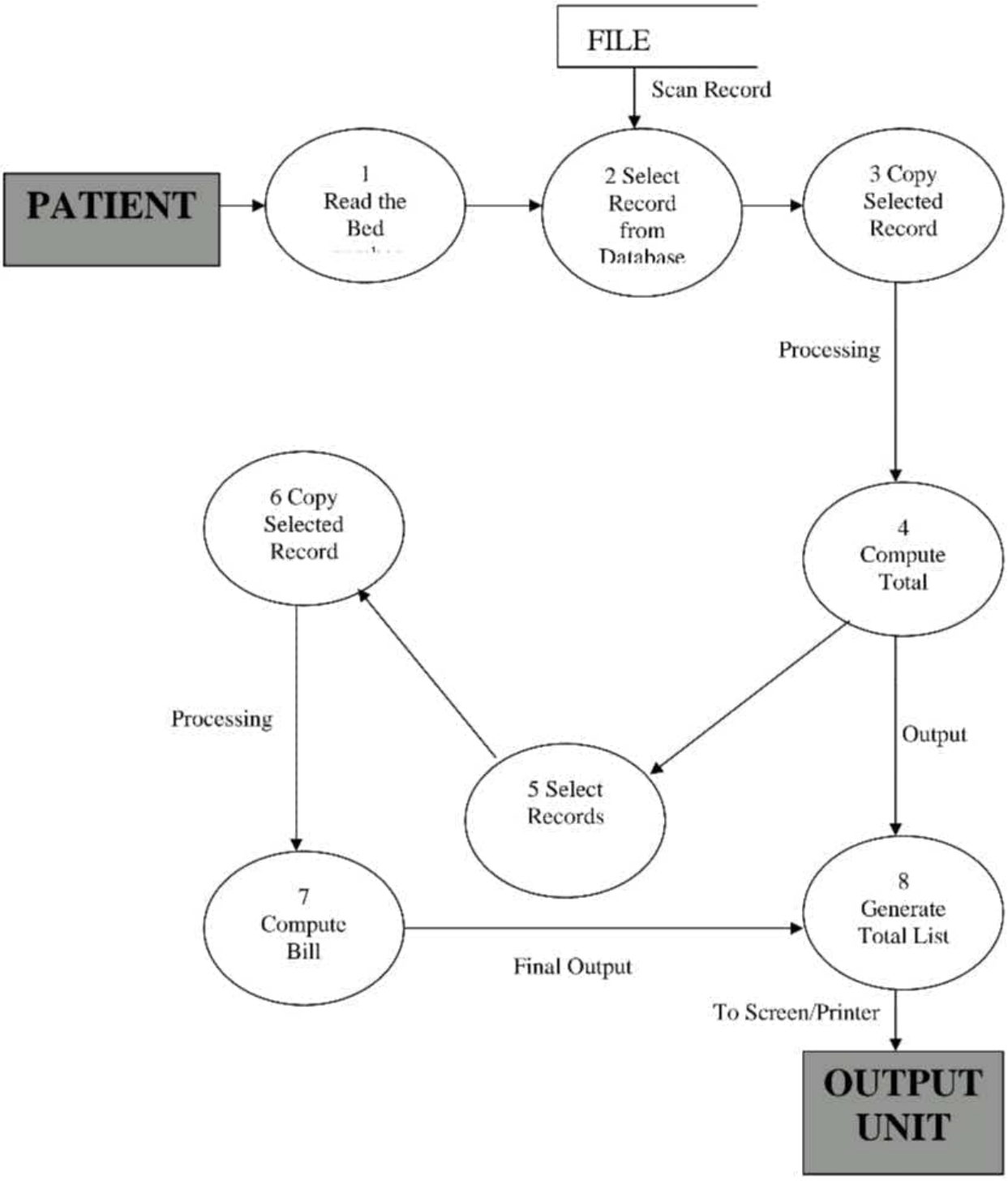
DATA FLOW DIAGRAM
RECORD MODIFICATION



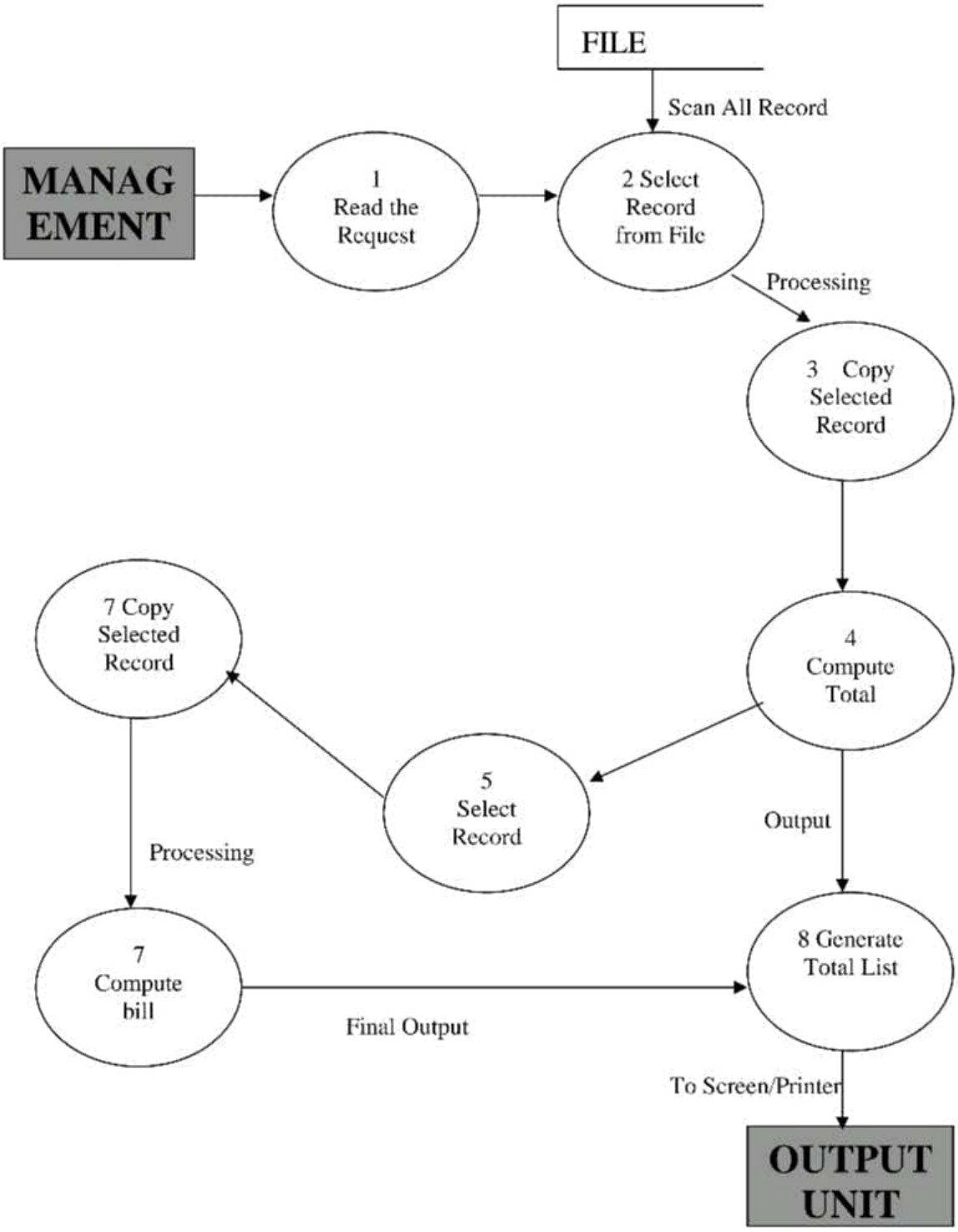
DATA FLOW DIAGRAM
DISCHARGE OF PATIENT



DATA FLOW DIAGRAM
LISTING OF PATIENTS



DATA FLOW DIAGRAM
LIST OF ALL RECORDS



SYSTEM DESIGN

System Design:

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -
Define the boundaries along which to break;
Determine into how money pieces to break; and

Identify the proper level of detail when design should stop and implementation should start. Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.
The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain. The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own

components. In a software system, a component is a software module. The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. A reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

A design can be object-oriented or function-oriented. In function-oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object-oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function-oriented methods for design and describe one particular methodology the structured design methodology in some detail. In a function- oriented design approach, a system is viewed as a transformation

function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other. Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other programmers. Generally, design specification uses textual structures, with design notation helping in understanding.

DATA MODELING:

Users table

Field	Type	Constraint
Name	Char (30)	Not Null
Emp Id	Char (30)	Primary Key
Email Id	Char (30)	Not Null
Password	Char (30)	Not Null

Admin

Field	Type	Constraint
Username	Char (30)	Not Null
Password	Char (30)	Not Null

Pateint table

Field	Type	Constraint
Card_no	Char (30)	Primary key
Name	Char (30)	Not Null
Gender	Char (30)	Not Null
Age	Numeric	Not Null
Address	Char (60)	Not Null
Phone	Numeric	Not null
Relative_name	Char (30)	Not null
Relative_address	Char(60)	Not null
Department	Char (60)	Not Null
Doctor_name	Char (30)	Not null

Doctor Master

Field	Type	Constraint
Dr_code	Char (30)	Not null
Dr_name	Char (30)	Not null
Gemder	Char (30)	Not null
Date_of_birth	Date	Not null
Address	Char (30)	Not null
Date_of_join	Date	Not null
Desgination		Not null

Bed_details

Field	Type	Constraint
Bed_no	Char(30)	Not null
Satus	Char(30)	Not null

OPD_master

Field	Type	Constraint
Name	Char (30)	Not Null
Card_no	Char (30)	Primary key
Patient_name	Char (30)	Not null
Gender	Char (30)	Not null
Age	Numeric	Not null
Address	Char(60)	Not null
Phone	Numeric	Not null
Rel_name	Char (30)	Not Null
Date	Date	Not null
Dr_unit	Char (30)	Not null
Days	Char(60)	Not null
Dep_name	Char (30)	Not null

Test_details

Field	Type	Constraint
Receipt_no	Char(30)	Primary key
Patient_id	Char (30)	Not null
Name	Char (30)	Not null
Date	Date	Not null
Report	Char (60)	Not null

Test_master

Field	Type	Constraint
Test_code	Char(30)	Not null
Test_test	Char(30)	Not null
Rate_per_test	Char (30)	Not null

Feedback

Field	Type	Constraint
Name	Char (30)	Not Null
Email Id	Char (30)	Not Null
Phone	Char (30)	Not Null
State	Char (30)	Not null
Comment	Char (60)	Not null

SCHEDULING:

Scheduling of a software project does not differ greatly from scheduling of any multi- task engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities.

Estimates of Effort

- A decomposition of the product function.
- The selection of the appropriate process model and task set.
- Decomposition of tasks.

Interdependencies among tasks may be defined using a task network. Tasks, sometimes called the project Work Breakdown Structure (WBS) are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models; and (3) calculate "boundary times" that define a time window" for a particular task.

Boundary time calculations can be very useful in software project scheduling. Slippage in the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (I) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.

Both PERT and CPM have been implemented in a wide variety of automated tools that are available for the personal computer. Such tools are easy to use and take the scheduling methods described previously available to every software project manager.

11. FURTHER SCOPE OF THE APPLICATION

1. Though maximum efforts have been put in to make this report authentic in all aspects and to take all necessary presentation to ensure that the information gathered is true, some uncomfortable factors may have crept in.
2. Some of the respondents were reluctant to part with certain information on the pretext of the sensitivity of the information. Also some facts of figures were not divulged as the company policy came in the way for free revelation of the desired input.
3. An element of bias might have crept in from the side of the official interviewed. This could also have resulted in some kind of modification of the information divulged.
4. Through an attemptwas make to collect information from the best possible source in the company, it was difficult to meet the top officials due to their busy schedules.
5. Most of the analysis and interpretations, made for this report, are based on secondary data obtained. This data could have some inherent mistakes and errors.
6. Finally, although due care has been taken those can be typing and compilation errors in the report itself.

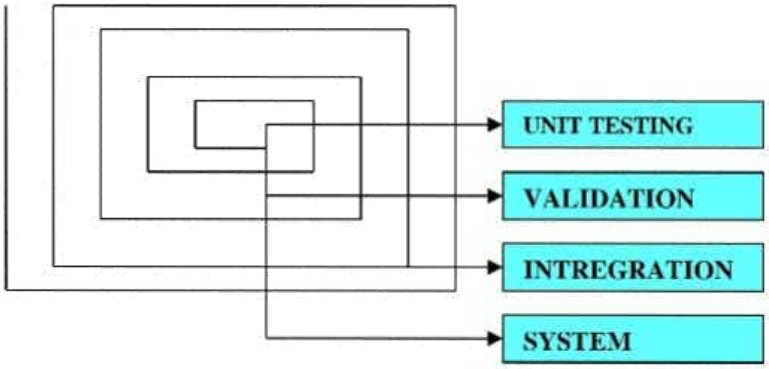
The tasks specified were not well defined because nothing was mentioned regarding validations in the project. Though we gave maximum effort to check the software with different validation tests, a few of them might be present in this version.

- Due to limited time available survey could not be undertaken for intended 20 consumers and thus had to be limited to 10.
- Communication gaps exist between employees and management, as seniors don't share problem with subordinates resulting in violation of psychological contract.
- Poor rewarding system(slow)
- Poor working conditions

The limitations may be many and the magnitude of the influence of these limiting factors may have a bearing on the report, but it in no way alters the ultimate aim of the project and because it's highly USER FRIENDLY, it would be the choice of all kinds of personnel.

LEVELS OF TESTING:

The different types of testing are as follows:



1. Unit Testing:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2. Integration Testing:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. System Testing:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

4. Field Testing:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned, we haven't tested our project using the field testing.

1. Acceptance Testing:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

Why System Testing?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

Another reason for system testing is its utility as a user-oriented vehicle before implementation.

Activity Network for System Testing

The test plan entails the following activities:

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile/assemble programs.
7. Prepare job performance aids.
8. Prepare operational documents.

Prepare Test

A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

- Outputs expected from the system.
- Criteria for evaluating outputs.
- A volume of test data.
- Procedure for using test data.
- Personnel and training requirements.

Specify Conditions for User Acceptance Testing

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

Prepare Test Data for Program Testing

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

Prepare Test Data for Transaction Path Testing

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

Plan User Training

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

Compile / Assemble Programs

All programs have to be compiled / assembled for testing.

Prepare Job Performance Aids

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

Prepare Operational Documents

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

Systems testing

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

Acceptance testing.

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the

responsibility of the programmer to collect the data that will produce the required test condition.

VERIFICATION AND VALIDATION (V&V):

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

The terms verification and validation are used synonymously. These are defined as under:

-

Verification: "Are we building the product right?"
Validation: "Are we building the right product?"

Verification activities include proving, testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities.

The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for each V&V task, and criteria for evaluating the outputs are also specified.

The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g. trace ability and document evaluation). Due to this, inspections play a significant role in verification.

DRAWBACKS OF CURRENT MANUAL- SYSTEM

1. The current manual system has a lot of paper work and it does not deal with old and new car purchase and sale.
2. To maintain the records of sale and service manually, is a Time-consuming job.
3. With the increase in database, it will become a massive job to maintain the database.
4. Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous details.
5. The retrieval of records of previously registered patients will be a tedious job.
6. Lack of security for the records, anyone disarrange the records of your system.
7. If someone want to check the details of the available doctors the previous system does not provide any necessary detail of this type.

ESTABLISH THE NEED OF NEW SYSTEM

1. Problem of Reliability: Current system is not reliable. It seems to vary in quality from one month to the, next. Sometimes it gives good output, but some times the output is worst.
2. Problem of Accuracy: There are too many mistakes in reports.
3. Problem of timeliness: In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.
4. Problem of Validity: The output and reports mostly contains misleading information. The customer's information is sometimes not valid.

5. Problem of Economy: The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired results.
6. Problem of Capacity: The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work.

PROPOSED SYSTEM

l. Employee Details: The new proposed system stores and maintains all the employees details.

2. Calculations: The new proposed system calculates salary and income tax automatically and it is very fast and accurate.

3. Registers: There is no need of keeping and maintaining salary and employee register manually. It remembers each and every record and we can get any report related to employee and salary at any time.

4. Speed: The new proposed system is very fast with 100% accuracy and saves time.

5. Manpower: The new proposed system needs less manpower. Less people can do the large work.

6. Efficiency: The new proposed systems complete the work of many salesperson in less time.

7. Past details: The new proposed system contains the details of every past doctor and patients for future assistance.

8. Reduces redundancy: The most important benefit of this system is that it reduces the redundancy of data within the data.

9. Work load: Reduces the work load of the data store by helping in easy updates of the products and providing them with the necessary details together with financial transactions management.

10. Easy statements: Month-end and day-end statement easily taken out without getting headaches on browsing through the day end statements.

NEED:

I have designed the given proposed system in the JSP to automate the process of day to day activities of Hospital like Room activities, Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc., online facilities to the multiple users etc.

The complete set of rules & procedures related to Hospital's day to day activities and generating report is called **"HOSPITAL MANAGEMENT SYSTEM".** My project gives a brief idea regarding automated Hospital activities.

The following steps that give the detailed information of the need of proposed system are:

Performance: During past several decades, the hospital management system is supposed to maintain manual handling of all the hospital daily activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the hospital management system, the computerized hospital management system is to be undertaken. The computerized hospital project is fully computerized and user friendly even that any of the hospital's members can see the patient's report and the doctor's report.

Efficiency: The basic need of the project is efficiency. The project should be efficient so that whenever a new patient is admitted, and automatically a bed is assigned and also a

BIBLIOGRAPHY

[1] Herbert Scheldt, **Java Complete Reference**, Fifth Edition, Tata McGraw Hill Edition.

[2] Phil Hanna, **JSP 2.0**: The Complete Reference, Tata McGraw Hill Edition, 2003.

[3] Elmarsi and Navathe, **Fundamentals of Database System** (Third Edition), Addision Wesley.

[4] Ian Somerville, **Software Engineering**, Third Edition, Pearson Education.

[5] Ali Bahrami, **Object-Oriented System Development**, Third Edition, Tata McGraw Hill Edition.

[6] Ivan Bayross, **SQL, PL/SQL programming language of Oracle,** Second Edition, BPB Publication.

WEB REFERENCES

[1] **www.google.com**

[2] **www.htmlcodetutorial.com**