

HOSPITAL MANAGEMENT SYSTEM

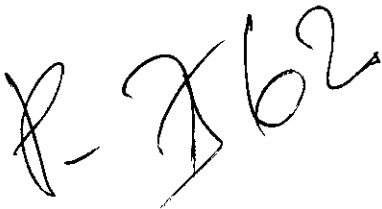
PROJECT WORK DONE AT

VOGUE SOFTWARE SERVICES LIMITED, CHENNAI

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF COMPUTER APPLICATIONS
OF BHARATHIAR UNIVERSITY, COIMBATORE.



SUBMITTED BY
D.MANIKANDAN
Reg. No. 9938M0618

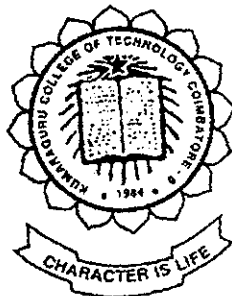
GUIDED BY

EXTERNAL GUIDE

MR. K.GOVIDHARAJAN, M.C.A.,

INTERNAL GUIDE

MR. N.S.RAMALINGAM M.C.A.,



Department Of Computer Science & Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

Department Of Computer Science & Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

Coimbatore-641006.

CERTIFICATE

This is to certify that the project work entitled

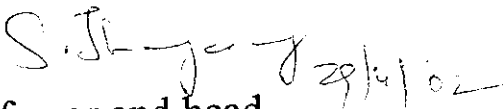
HOSPITAL MANAGEMENT SYSTEM

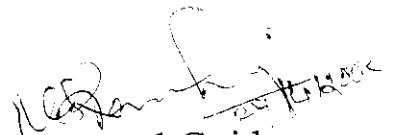
Done by

D.Manikandan
9938M0618

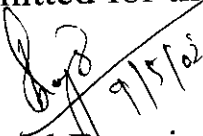
Submitted in partial fulfillment of the requirements for the award of the
degree of

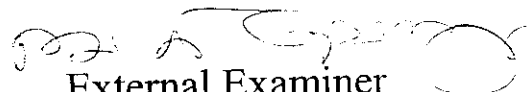
Master of Computer Applications of Bharathiar University.


Professor and head


Internal Guide

Submitted for the university examination held on 09.05.2002


Internal Examiner


External Examiner

VOGUE

SOFTWARE SERVICES

J10, Baid Mehta Complex,
183, Mount Road,
Little Mount,
Chennai – 600 015
Phone: 044 – 2200665

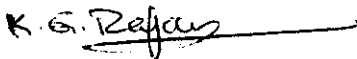
Date : 22-04-2002

CERTIFICATE

This is to certify that Mr. D.Manikandan (99MCA24) has been successfully completed the project work on HOSPITAL MANAGEMENT SYSTEM during the period of Dec'2001 and Mar'2002.

We wish him all the best in his future endeavors.

Thanking you




Govindarajan
(General Manager)

DECLARATION

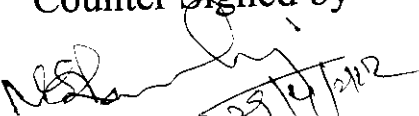
I here by declare that the project entitle "**HOSPITAL MANAGEMENT SYSTEM**" submitted to **Bharathiar University** as the project work of Master Of Computer Applications Degree, is a record of original work done my me under the supervision and guidance of **Mr.K.Govindarajan** and **VOGUE SOFTWARE SERVICES LTD., Chennai** and this project work has not found the basis for the award of any Degree /Diploma/ Associateship/ Fellowship or similar title to any candidate of any university.

Place: COIMBATORE

Date : 29.02.2002.


Signature of the student

Counter Signed by


(Internal guide)

Acknowledgment

ACKNOWLEDGMENT

I express my profound respect and sincere gratitude to **Dr.K.K.Padmanabhan Ph.D**, Principal, Kumaraguru College of Technology, Coimbatore, for his kind co-operation in allowing me to take up this project work.

I record my sincere thanks to **Dr. S. Thangasamy Ph.D**, Head of the Department, Computer Science and Engineering, Kumaraguru College of Technology, for allowing me to take up the project at vogue software services Limited., Chennai.

I wish to extend my gratitude to **Mr. K. Govindharajan, MCA**, Project Manager , vogue software services Chennai, for allowing me to carry out the project at their organization and guiding me in completing the project successfully.

I am greatly privileged to express my deep gratitude to my guide **Mr.N.S.Ramalingam, MCA**, Dept of Computer Science and Engineering, Kumaraguru College of Technology, for his valuable advice and encouragement.

I also owe my sincere thanks to **Mrs. Geetha Vellingiri, MCA**, Course coordinator, Master of Computer Applications, Kumaraguru College of Technology, Coimbatore for her guidance and immense support throughout my project work.

I also take immense pleasure in thanking everyone who were directly or indirectly involved in the success of this project.

D.Manikandan

Synopsis

SYNOPSIS

The project entitled "*Hospital Management System*" is developed for Vogue software services limited.

The project is done for a hospital which has fields for collecting information on patient's detail, Doctor's detail, laboratory and Operation details, Payment of the patient, consultation history, consultant doctor for the patient, nature of ailment, medical tests done and cure suggested. The information is added using data input templates.

When the patient enters the hospital for treatment. The administrator checks the patient whether the patient is new patient or registered patient. If the patient is the registered patient the administrator collects the patient detail by giving the patient-id. The output form displays the details.

If the patient is a new patient the administrator creates a new patient-id for the patient and records all the information about the patient. This details is used by the doctor whenever the need arises.

If the doctor decides the patient has to be admitted, the patient is categorized as in-patient or outpatient. If the patient is considered as an inpatient the administrator records the information about the patient.

The bed allocation table gives the information about the availability of beds in the hospital. From this table the bed is allocated to the patient. The doctor starts treatment to the patient from the result obtained from the lab. The tests and their results are also recorded in the in-patient file.

If the patient is outpatient the administrator creates the table for that patient. The information contains patient's number of follow-ups, frequency of follow-ups and last visit.

The system contains the details about the doctors. The doctors are categorized as in-house doctor or visiting doctor. If the doctor is an in-house doctor the table contains the details about the doctor, specialization of the doctor and contact numbers are recorded.

If the doctor is a visiting doctor the administrator creates a doctor code and enters the detail about the visiting doctor. It contains the detail about the visiting days,

The operation theater table contains the patient-id, type of operation, doctor code, date of the operation and the cost for the operation. If the doctor decides to discharge a patient the pay list is prepared for that patient. The bed is automatically freed in the bed allocation table. The pay list produces the total bill of the patient.

The project is done with front-end tool as ASP and the back-end tool as MS SQL server 7.0. Windows Me is the operating system.

CONTENTS

	Page No
1. Introduction	
1.1 Project Overview	01
1.2 Organization Profile	04
2. System Study & Analysis	
2.1 Existing System - Limitation	05
2.2 Proposed System.	06
2.3 User Characteristics	07
2.4 Requirements on New System	08
3. Programming Environment	
3.1 Hardware Configuration	09
3.2 Software Requirements	09
4. System Design & Development	
4.1 Input Design	13
4.2 Output Design	14
4.3 Database Design	15
4.4 Process Design	16
5. System Implementation & Testing	
5.1 System Implementation	21
5.2 System Testing	22
6. Security	26
7. Conclusion	27
8. Scope for future development	28

Appendix

- A. Sample Screens
- B. Sample source code
- C. Tables

Introduction

1. INTRODUCTION

1.1 Project Overview

The document contains a general description of product perspective, product function user characteristics & general constraints. It also has detail about function requirement, performance and design constraints

The project has fields for collecting information on patient details

- Patients diagnosis
- Patients medical history
- Suggested Treatment
- In-house Patients Details
- Out Patients Medical Treatment
- Payment Details
- Bed Allocation.
- A web based Physician-Patient Consultation

Patients Diagnosis

The administrator collects series symptoms. The Symptoms, which are collected separately and forwarded to the doctor. Doctor starts the treatment keeping the symptoms as the base.

Patients Medical History

The administrator checks the patient whether he is a new patient or registered patient. If the patient is registered patient the administrator gathers the information, which already exists in the database. If the patient is new then he creates a new patient-id after entering the personal details. Previous records of the patients are analyzed for the further treatment.

In-House Patients Details

If the doctor decides a patient as inpatient then the patient details such as name, blood group, disease found, etc, are entered through a form which is stored in the database and updated when necessary. Medicines for the In-House Patients are prescribed and also entered in to the database.

Outpatients Details

If the patient is found to be an outpatient the administrator stores the detail of the patient and medicines are recorded. This record is further updated when needed. The administrator enters the patients history as per the doctors diagnosis

Suggested Treatment

The doctor enters the patient-id, the system will generate the patients report. The doctor then Analyses the previous records and suggests for treatment to be taken.

Payment Details

Billing is made when the user enters the patient-id. The system will automatically search the entire expenses of the patient from the treatment given and produces this summary as a bill. The expenses includes

- Lab expenses
- OT expenses
- Medicine expenses
- Doctor consultation fee.
- Room rent (if in-patient)

Bed allocation

The system keeps track of the beds available in the hospital. Each bed is checked for occupancy, if the bed is occupied then the bed is associated with the patient-id. This will allow the system to check for the availability of unoccupied bed when a new patient enters.

Web based Physician-Patient consultation.

The web-based chat program will allow the patient to talk to the doctor after he is discharged. This chat program enables the patients to create password through which he enters into the program. This chat program is available with the hospitals website. This module is fully controlled by the administrator and he maintains the Timing details of availability of the doctor.

1.2 About the Organization

Vogue software Services was formed in 1998 in the state of Tamil Nadu, India. Vogue is an international staffing organization that provides personnel to Fortune 500 corporations worldwide. Our clients include members of the financial, insurance, civil engineering industries and hospital management.

Vogue Software Services - a Solution Provider

- At the technology center in Chennai, a dedicated team of IT professionals is put into vigorous clients specific training programs.
- The management of Vogue Software Services has assembled a team of professionals skilled in the delivery of our core competencies, which includes consulting services, software development and product development.
- Vogue Software Services is a guaranteed low cost provider.
- Vogue Software Services is a growing organization, offering flexibility to do what it takes to provide IT staffing solutions that fit its client's needs.

*System Study
& Analysis*

2. SYSTEM STUDY & ANALYSIS

2.1 Existing System

The existing system is developed in VB, this existing system work as a Single User system and it is not available online.

Limitations of Existing System

- The Existing system has no Accessing capability of all doctors. The Accessing capability allows network administrators to explicitly disallow access to any "unacceptable" persons.
- Caching of Web Pages is not done in Existing system.
- The system was not secure. As it is not password protected.
- The system was not able to accommodate large amount of data as the patient increased.
- Timely reports and bills were not produced.
- The system was designed for batch processing which will lead to greater errors and inefficiencies.
- There is no Web Based consultation between Doctor and the Patient.
As the system was a stand-alone system, the users were not interactive and only one person was allowed to enter the details. Therefore if that person is absent or on leave the whole system will stop.

2.2 Proposed System

Information which is the backbone of a organization has to be made available at all times to ensure proper decision making, and moreover it has to be timely, accurate, relevant and usable.

- The administrator collects series symptoms. The Symptoms, which are given are collected separately and forwarded to the doctor. Doctor starts the treatment keeping the symptoms as the base.
- Previous records of the patients are analyzed for the further treatment.
- Analyzing the previous records the doctor suggests for treatment to be taken.
- Medicines for the in-house Patients are prescribed And it is updated.
- Details of the patient and medicines are recorded.
- This module is fully controlled by the administrator and He maintains the Timing details of availability of the doctor.

2.3 User characteristics

Users of this software are classified into four levels. They are

- Higher authorities
- Doctors
- System administrator
- Data entry operator

Data entry operators are the lower level user they have only read permission such as taking printouts. He will be trained before using the product.

The system administrator has the rights to give the input to the system.

The doctors of the hospital have the rights of all their lower level subordinates have and their own rights. They can be able to write all the decision-making information into the system.

Higher authorities have the entire read, write and delete permissions.

The data entry operator should know about the hospital functioning.

2.4 Requirement of the new system

The new system will need training for the users, which will let them to use the system more efficiently.

- The new system will need intranet facilities
- It require the web server
- The system should be user friendly and interactive
- It should have a speedy fulfillment and tracking system. The user can achieve the Updating and retrieval of data at any time.

*Programming
Environment*

3. PROGRAMMING ENVIRONMENT

3.1 Hardware Configuration

The Hospital Management system is developed using the computer infinity model, whose specification is as follows:

Intel Pentium III

20GB Hard Disk

64MB RAM

Cache Memory 512 KB

Network Interface Card

3.2 Software Requirements

OPERATING SYSTEM: Windows Me/NT

Internet Explorer 5

The system is developed using

- **PWS (PERSONAL WEB SERVER)**

A web server used to satisfy the client requests.

- **ASP (ACTIVE SERVER PAGES)**

An advanced tool to communicate with the database and server by writing scripts.

- **HTML (HYPERTEXT MARKUP LANGUAGE)**

Tool to provide good interface for use in web pages.

- **JAVA SCRIPT AND VB SCRIPT**

Scripting language used to process client side requests faster.

- **MICROSOFT SOL SERVER**

Back end database to store data.

Description of Software & Tools Used

After a thorough analysis, ASP has been selected for building this application. This application runs under Windows platform.

- The system is highly adaptable to the web and local intranet; therefore the system is coded with Active Server Pages-a Microsoft product.
- The server side scripting is done in VBScript and the client side chatting is done in JavaScript.
- SQL Server is chosen as the back-end as it is highly secure

The Active Server Page Environment

Generate dynamic web pages. An ASP can display different content to different user.

Process the contents of the HTML forms.

Create database-driven Web pages.

Track user. With ASP you can limit the access of certain Web pages to certain individuals.

Integrate custom components into the Server. You can extend your ASP scripts with custom server-side components created with Microsoft script components, Microsoft VB, or Microsoft VC++.

Reasons for selecting Active Server Pages

The ASP is a technology for building dynamic and interactive web applications. The primary difference between ASP and the other new generation technologies mentioned is that Asp must be executed on the web server, while the pages generated by

the browser or client). And the advantages that ASP, which makes us to choose our project, is that,

- **Simplicity and speed:** It is very simple and speed over the CGI and Pearl.
- **Browser Independence:** The ASP portion of a page runs on the server and sends only results to the client independent of the browser that is used by the clients.
- **Active Server Objects:** There are six Active Server Objects, each of which deals with specific aspect of interactivity and are intended and are intended to simplify browser and page control

The SQL Server Environment

It is the widely used database and runs on virtually any kind of computer.

SQL Server is an object-relational database. An object relational database supports all the features of a relational database while also supporting object-oriented concepts and features.

SQL Server is ODBC complaint.

Reasons for selecting SQL Server

SQL Server is a powerful multi-user RDBMS, which is used in a project as a backend to store data.

SQL Server has been chosen within the system for following reasons.

- Dynamic self-management of things like memory and locking
- Very large database support
- Better performance

- Better integration with Windows NT security
- More flexible replication
- Full Unicode support
- Centralized management for multiple sites
- Data warehousing and OLAP support
- Microsoft Management Console integration
- Improved Data Transformation Services
- Row level locking
- Parallel query processing
- Distributed query processing
- Many new Wizards

INTERNET INFORMATION SERVER (IIS)

The Microsoft IIS software is included with the Windows NT 4.0. We are using the Microsoft's Internet Information Server (IIS) because for the reason to publish the web applications into Intranet. IIS combines Web, FTP and Gopher services and are managed by a single application, the Internet Service Manager.

The Microsoft Internet Information Server (IIS) is a secure, high-speed information publishing system that runs on Windows NT server. IIS is highly extensible through the use of Asp.

*System Design &
Development*

4. SYSTEM DESIGN & DEVELOPMENT

4.1 Input Design

The input design of the system is user friendly and graphical user interface (GUI) enabled. The forms are designed in such a way that the user does minimum typing.

The system uses Variety of controls like

Radio buttons

Dropdown list

Text area

Text box

Submit and reset buttons

The radio buttons are used to select the sex fields. The dropdown list used to select the fields like marital status, type of patient and doctor detail. The text area is used to give input for the fields like address, test, result and patient history. The name, age, phone number fields are entered through text box. The submit buttons are used to post the input page for validation and processing.

The error messages are customized; therefore the user understands the error and correction is made simpler.

Initially the users should give the following inputs:

User name

Password

If the user is found to be valid, the system displays the options page. This page

- New patient
- Registered patient.

4.2 Output Design

The data stored in the database and the data that is to be collected from the user, needs specific output layouts, which is user friendly and understandable even by a novice.

The basic system has output layouts, namely

- Reports

The output screen displays the details about the patient by giving the patient-id. If we select the doctor code in the doctor detail code the form will display the doctors detail.

The payment table produces the report by fetching the details from the different tables in the system. The data entry operator then makes the displayed details in to hard copy.

- Since Hospital Management System is an on-line system, enquires are made regarding the tests, patient details.
- Reports are of standard as well as ad-hoc type. Some of the reports are patient detail, which gives the list of patients and their details.
- Payment report, which give the report of a patients payment list.
- The form shows the reports
- In the case with the services for all the master entries, modification and deletions confirmation message accordingly is given

There are several reports to the users and the administrator like reports on registration, reports on lab, reports on operation, reports on payment and several patient reports like number of registrations, number of tests, number of payment etc., In general all the input screens are give with a confirmation message which will be very interactive for the users of the software.

4.3 Database Design

The system database is designed in keeping the following in mind

Ease of use

If the user wants to display a report, just an entry of patient-id is enough to fetch patient's detail. The user need not be bother about the database structure as the system takes care of it.

Data independence

The user is given the option of omitting curtain unnecessary fields. The database design such a way that the system allows the user to skip certain part of the entry like fax, mail-id, etc.

Accuracy and integrity

Data entered by the user in the form as to be stored in the database in the same format and if there are relational tables the data has to reside in the entire table.

Recovery from failure

The database is designed for incremental backups so that any system failure does not lead to havoc.

Performance

The database is designed to retrieve information with maximum speed. This is done through normalization. The database can adapt to any no of records.

Redundancy

The replication of data is eliminated by improved database design. The redundancy of fields needs extra storage space, hence redundancy is avoided.

4.4 Process Design

Once the input, output and storage have been designed the next step is to design the process that acts upon the data storage to give the required output. The process for hospital management was chosen based on the following factors, they are

Response time

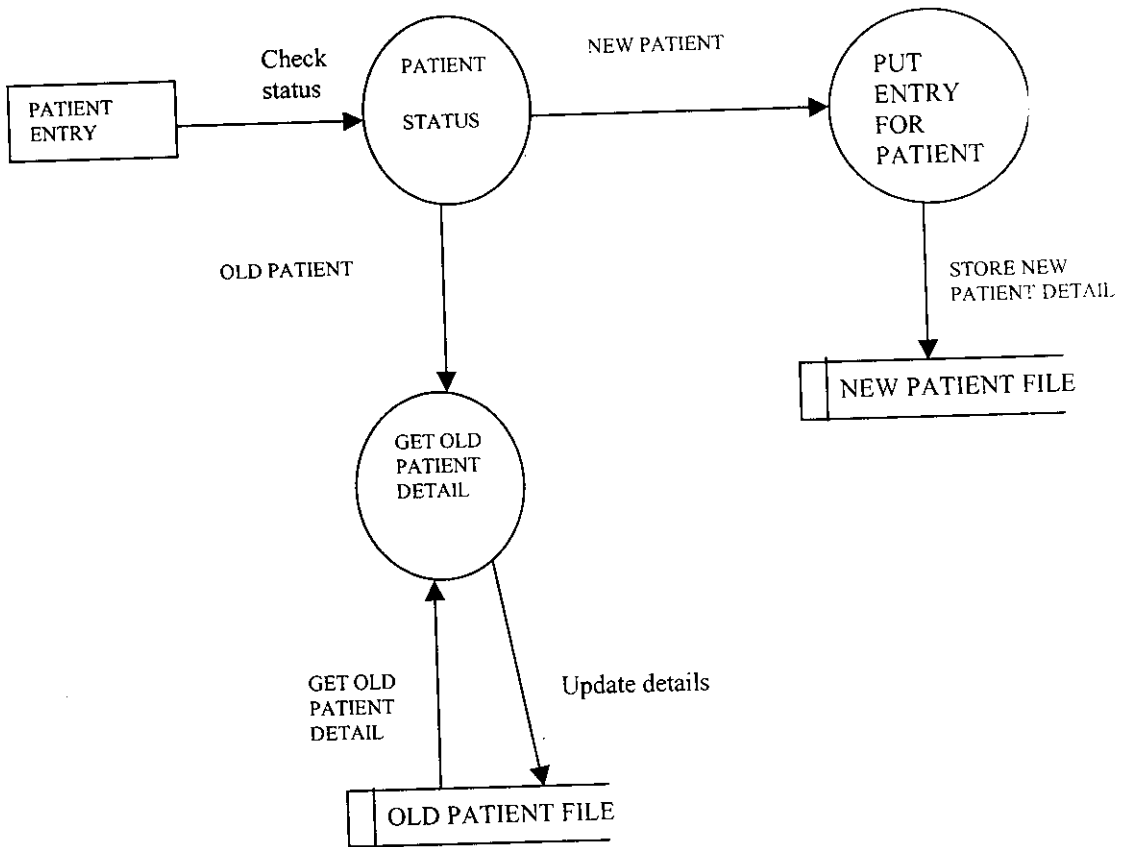
Data volume

Hardware constrains

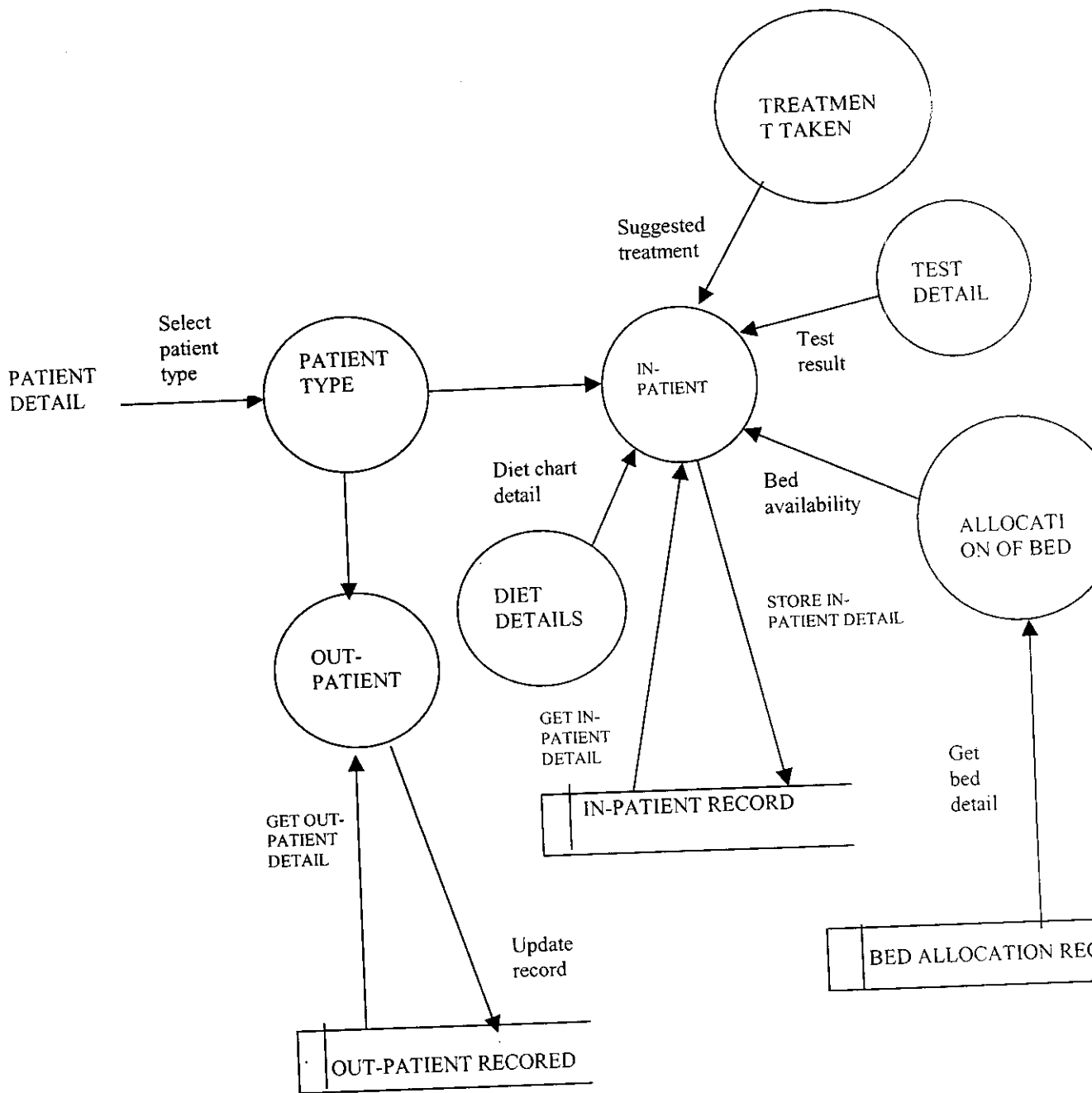
Security factors

The process design can be able to handle huge volume of data and it also designed suitable for client/server architecture so that it will have higher degree of performance and speed. The process design of our project is given as a data flow diagram.

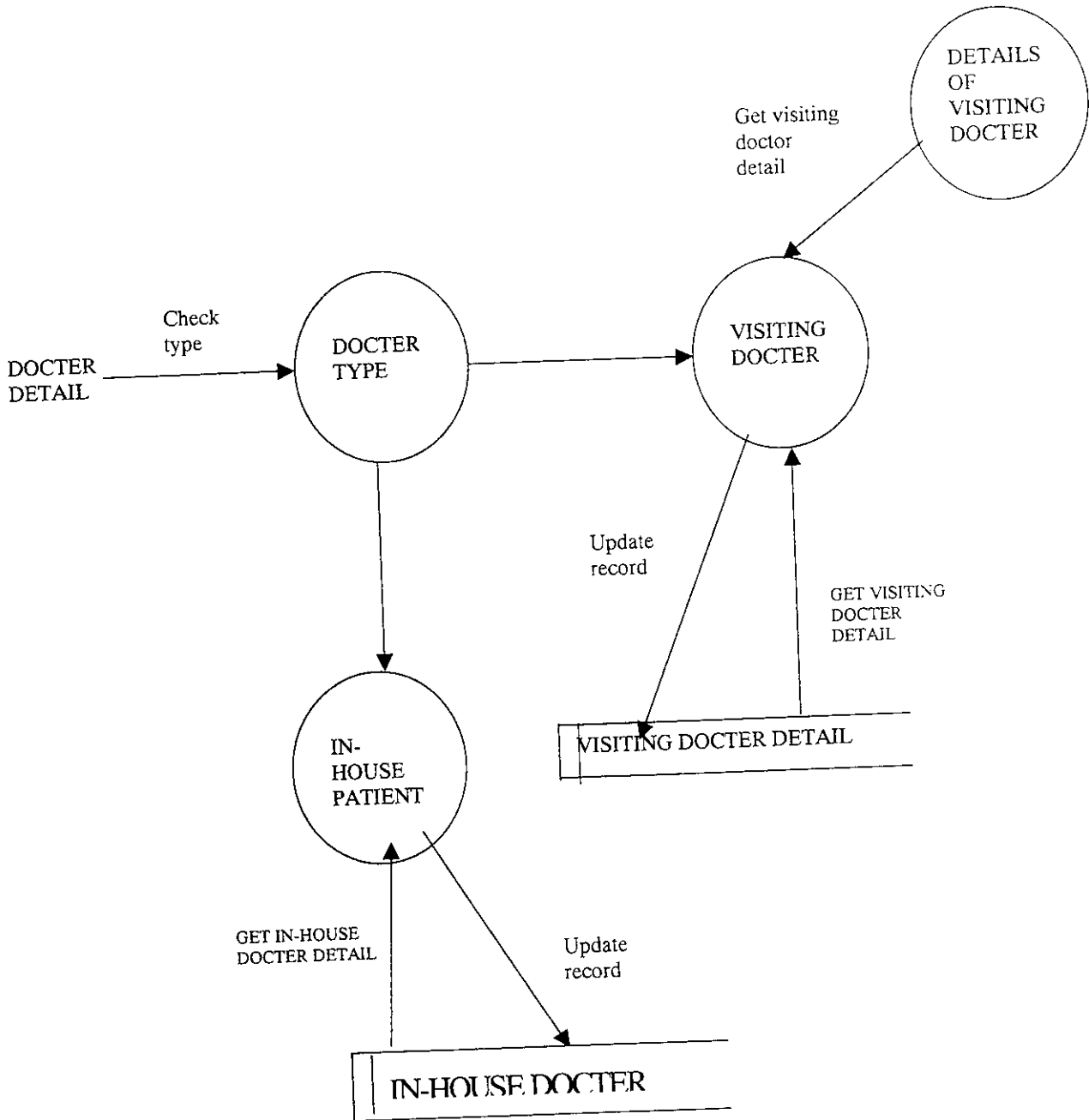
PATIENTS PERSONAL DETAIL



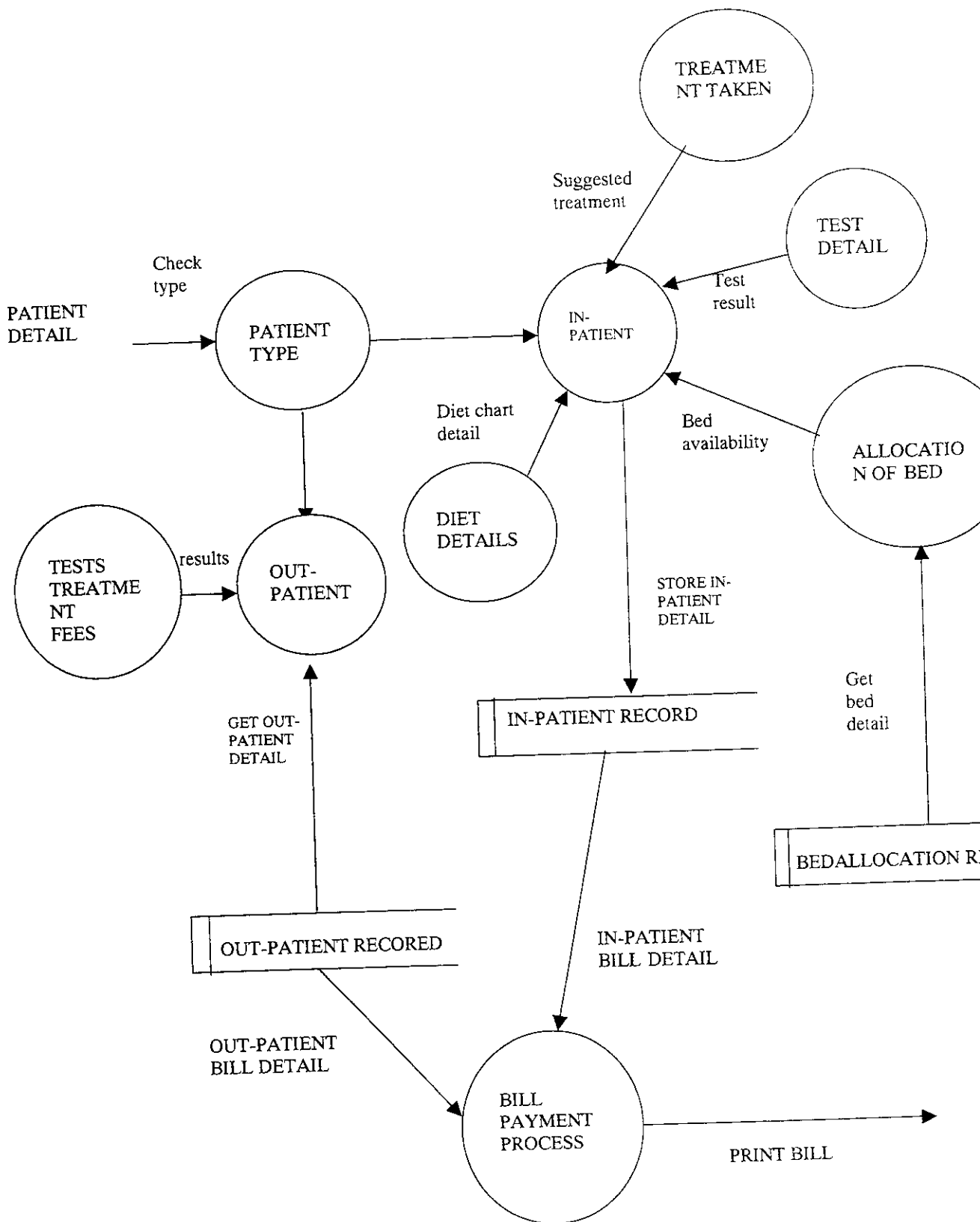
IN-PATIENT AND OUT-PATIENT



DOCTERS DETAIL



PAYMENT DETAIL



***System Implementation
& Testing***

5. SYSTEM IMPLEMENTATION AND TESTING

5.1 System Implementation

Implementation is the stage of the project when the theoretical design is turned into a working system. At the stage the main workload, the greatest upheaval and the major impact on existing practices shift to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. Thus it can be considered to be the most crucial stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective.

The implementation stage is a system project in its own right. It involves careful planning, investigation of current system and its constraints on implementation, design of method to achieve the changeover, training of staff in the changeover procedure and evaluation of changeover methods.

The main task of implementation are classified as follows

Implementation planning

Computer system testing

Education and Training

Implementation Planning

The implementation of a system involves people from different department and we are confronted with the particular problem of controlling the activities of people outside their own data department. Because of this we have to plan carefully for accessing the respective staff members. This involves meeting the respective staff members only with the permission of their managers and without disturbing their usual routines. And our implementation process should not disturb or collapse the existing system.

5.2 System Testing

Testing Process

Except for small software, systems should not be tested as a single, monolithic unit. Large systems are built out of sub-systems, which are built out of sub-systems, which are composed of procedures and functions. The testing process should therefore proceed in stages where testing is carried out incrementally in conjunction with the system implementation.

There are the five test stages and defects are discovered at any stage, they require program modifications to correct them and this may require other stages in the testing process to be repeated. The process therefore is an iterative one with information being fed back from later stages to earlier parts of the process.

The stages in the testing process are:

Unit Testing

Module Testing

Sub-system Testing

System Testing

Acceptance testing

Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. With respect to this project, the individual functions are treated as component and were tested.

Module Testing

A module is a collection of dependent components such as an object class, an abstract data type or some looser collection of procedures and functions. A module encapsulates related components so it can be tested with other system components.

Sub-system Testing

This phase involves testing collection of modules, which have been integrated into sub-systems. Sub-systems may be independently designed and implemented. The most common problems that arise in large software system are sub-system interface mismatches. The sub-system test process should therefore concentrate on the detection of interface errors by rigorously exercising these interfaces. Both the modules are treated as a sub-system and tested in this stage.

System Testing

The sub-systems are integrated to make up the entire system. The testing process is concerned with finding errors, which result from unanticipated interactions between sub-system and system components. It is also concerned with validating that the system meets its functional and non-functional. After integration of the above sub-systems with the whole system, the entire system is tested for errors.

Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system procurer rather than simulated data. Acceptance testing may reveal errors and omissions in the system requirements definition because the real data exercises the system in different ways from testing the data. Acceptance testing may also reveal requirement problems where the system facilities do not really meet the users need or the system performance is unacceptable. Test and reveal data were provided to the system and checked for errors

Defect Testing

Defect testing is intended to exercise a system so that latent defects are exposed before the system is delivered. This contrasts with validation testing which is intended to demonstrate that the system meets its requirements. Validation testing requires the system to perform correctly using given acceptance test cases. A successful defect test is a test, which causes a system to perform incorrectly and hence exposes the defects. It demonstrates the presence, not absence of program faults.

Various values, within the limit and exceeding the limit were provided repeatedly to individual components of data acquisition. These brought out the defects in the system and were corrected.

Black-box Testing

It relies on the specification of the system or component, which is being tested to derive test cases. The system is 'black-box' whose behavior can only be determined by studying its inputs and the related outputs. This is also called as functional testing because mathematical functions can be specified using only inputs and outputs.

Following black-box methods were applied to both the modules to test arrays:

- Usage of only one value of entire array. This proved that the program works for an exceptional array.
- Usage of different arrays of different sizes. This decreased the chances that the program with defect would accidentally produce a correct output because of some characteristic of the inputs.
- First, middle and last elements were accessed and any problems due to the boundary effects were delivered.

Structural Testing

This is the complementary approach to black box testing and is sometimes called structural, white-box or glass-box testing. The tester can analyze the code and the use knowledge about the structure of the component to derive test data.

The advantage of structural testing is that an analysis of the code can be used to find how many test cases are needed to guarantee a given level of test coverage. A dynamic analyzer can then be used to measure the extent of this coverage and help with test case design.

Path Testing

Path testing is a white-box testing strategy whose objective is to exercise every independent execution path through the component. If every independent path is executed then all the statements in the program must have been executed at least once. Furthermore, all conditional statements are tested for both true and false cases. This helps to improve the program efficiency with respect to time complexity and memory usage.

6. SECURITY

- The system authorizer's user with password. Only the user with valid password can operate the system.
- Mechanism should be provided so that only permitted users are only eligible for modifying the table.

The potential threats are:

- Errors and omissions
- Disgruntled and dishonest employees
- Fire, natural disaster, external attacks.

SYSTEM SECURITY MEASURES

After the system security risks have been evaluated, the next step is to select security measures.

The measures are:

- Identification
- This is accomplished through password or devices such as SMART cards, etc. These help to identify if the user is a authorized person.

Conclusion

7. CONCLUSION

The Hospital Management System has been developed for the present requirements.

On-line information gives the Doctors flexibility to act decisively and in time. The developed system has to a good extent succeed in rectifying the present problems. The system can be further extended to host great features that are not currently included.

The newly developed system consumes less processing time and hence increases the throughput considerably.

*Scope for future
Development*

8. SCOPE OF FUTURE DEVELOPMENT

The additional enhancement for the product can be done in following ways,

- Tables should be created in all RDBMS.
- More templates can be included.
- Automatic generation of code for Data Entry forms.
- Providing access cards can enhance the security feature.
- Dynamic field allocation in the tables can be added.

Bibliography

BIBLIOGRAPHY

Reference Books

- ✓ Stephen Walther with Jonathan Levine “Teach yourself E-Commerce in 21 days” - Sams Publications - 2001

- ✓ A.Russell Jones “ Mastering Active Server Pages 3.0”
BPB Publications - 2000

- ✓ Mark Baarsle “Professional ASP with XML”
Wrox Publications – 2000

- ✓ James Jaworski “Mastering JavaScript and Jscript”
BPB Publications - 2000

- ✓ Mike Gunderlog and Maryehipman “SQL Server 7.0”
BPB Publications - 1999

- ✓ Lee “Structured System Analysis and Design”
Galgotia Publications - 1980

Websites

www.aspdeveloper.net

www.15seconds.com

www.asptoday.com

www.versign.com

Username:

Password:

Login

New user? [Sign up now!](#)

- About us
- Doctors detail
- Visiting doctor detail
- Patient information
- Lab detail
- Emergency
- Out-patient Detail
- In-patient Detail
- Operation Detail
- Online chat

New Patient

Registered Patient

Hospital Path Way Specifications - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address <http://mani/mani/impat.asp>

Go Links

	inp12
	4/23/2002 11:50:03 AM
	doc23
	blood test,xray,scan
	O+ve,broken found
	fracture

Done

Local intranet

Start Misc... Ho... project Micr...

11:54 AM

View & Modify in-patient Details

choice	
choice	
ip1	
ip2	
ip3	
inp11	
e1	1:56:46 AM
e1	
inp12	

blood	▲
test, xray, scan	▼

Hospital Path Way Specifications - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail Stop

Address <http://mani/mani/inpat.asp> Go Links

```
patient_id=inp12
date of entry=4/23/2002 11:50:03 AM
doctor_code=doc23
disease=fracture
test=blood test,xray,scan
result=O+ve,broken found
medicines=given
diet_code=dc_B
room_type=class_C
bed_no=23
```

Record Updated!

Local intranet

Done Start Misc... Ho... project Mic...

11:55 AM

Address http://mani/mani/doc.asp

12, khkhgskg, asdsad, a
sdd

044-234567

098441-23415

mani@rediffmail.com

2345678

Permanent

Permanent

Visiting

[Update Doctors Detail](#)

[Go To Reception](#)

submit clear

Done

Local intranet

Visiting Doctors Details

	doc12
	raj
	GH
	dermatologist
	<input type="checkbox"/> Mon <input checked="" type="checkbox"/> Tue <input type="checkbox"/> Wed <input checked="" type="checkbox"/> Thu <input type="checkbox"/> Fri <input type="checkbox"/> Sat <input type="checkbox"/> Sun
	5-6pm 10am
	23, fdghfghgfh
	3242342

Lab Details

	inp12
	x-12
	ravi
	X-ray
	150.00

View & Modify Lab Details

	choice
	choice
	p2
	khghjg
	khghgkj
	khgkjhgkjhg
	3453.9

<< >> Save Delete

Local intranet

View & Modify Emergency Patient

	Choice
	Choice
	e1
	e11
	34
	ed5
	heart attack
	asdassdard

Hospital Management - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Refresh Stop

Address <http://mani/mani/emer.asp> Go Links >>

```
patient_id=e11
Age=34
doctor_code=ed5
disease=heart attack
treatment=asdassdasd
b_group=A1-ve
doe=4/23/2002 12:11:05 PM
remark=-
```

Record Updated!

Emergency

Local intranet

Done Start Misc... Ho... project Mic...

12:12 PM

Operation theater Details

	out33
	OT-4
	raja
	fracture
	mani
	4/23/2002 12:07:38 PM
	24000.00

submit clear

B. Sample source code

The sample source code for the table patient information is given below

```
<%@ language=VBScript%>
<% option explicit %>
<HTML>
<HEAD>
<TITLE> doc.htm </TITLE>
</HEAD>
<BODY bgColor=Red>
<!-- #include file = "connection.asp" -->
<%
dim d_code
dim name
dim specialisation
dim sex
dim address
dim phone
dim mobile
dim email
dim fax
dim d_type
dim i
dim errmsg
errmsg=vbnullstring
dim rsmember

set rsmember = server.createObject("ADODB.Recordset")
rsmember.cursorType = adOpenKeyset
```

```

rsmember.LockType = adLockOptimistic
rsmember.Open "Doc1",conn,,adCmdTable
if Request.ServerVariables("REQUEST_METHOD")="POST" then
    d_code=Request.Form("d_code")
    name=Request.Form("name")
    Specialisation=Request.Form("Sp")
    sex=Request.Form("Sex")
    address=Request.Form("Address")
    phone=Request.Form("phone")
    mobile=Request.Form("Mobile")
    email=Request.Form("mail_id")
    fax=Request.Form("Fax")
    d_type=Request.Form("d_type")
    d_code=trim(d_code)
    name=trim(Name)
    Specialisation=trim(Specialisation)
    sex=trim(Sex)
    address=trim(Address)
    phone=trim(Phone)
    mobile=trim(Mobile)
    email=trim(Email)
    fax=trim(Fax)
    if len(d_code) = 0 then
        errMsg = "You must enter Doctor code."
    else
        for i = 1 to len(d_code)
            if instr(1, "abcdefghijklmnopqrstuvwxyz1234567890", _
                Request.Form("d_code") & i) < 1 then
                errMsg = errMsg & "Invalid character: " & Request.Form("d_code") & i & " "
            end if
        next i
    end if
end if

```



```

        "invalid. Please re-enter doctor code."
    exit for
end if
next
end if
if errmsg <> 0 then
if len(name) = 0 then
    errMsg = "You must enter your name."
else
    for i = 1 to len(name)
        if instr(1, "abcdefghijklmnopqrstuvwxy", _
            mid(name,i,1), vbTextCompare) = 0 then
            errMsg="The name you entered is " & _
                "invalid. Please re-enter your name."
            exit for
        end if
    next
end if
end if
if len(Specialisation) = 0 then
    errMsg = "You must enter Specialisation."
else
    for i = 1 to len(specialisation)
        if instr(1, "abcdefghijklmnopqrstuvwxy", _
            mid(Specialisation,i,1), vbTextCompare) = 0
then
            errMsg="The Specialisation you entered is " &

```

```

        exit for
    end if
next
end if

if len(address) = 0 then
    errMsg = "You must enter patients address."
else
    for i = 1 to len(address)
        if instr(1, "abcdefghijklmnopqrstuvwxyz", _
            mid(address,i,1), vbTextCompare) = 0 then
            errMsg="The Address you entered is " & _
                "invalid. Please re-enter patient address."
            exit for
        end if
    next
end if

Response.Write len(errMsg)
if len(errMsg) = 0 then
    ' no validation errors occurred
    response.write "d_code=" & d_code & "<br>"
    Response.Write "Name=" & name & "<br>"
    Response.Write "Specialisation=" &
Specialisation & "<br>"
    Response.Write "Sex=" & sex & "<br>"
    Response.Write "Address=" & address & "<br>"
    Response.Write "Phone=" & phone & "<br>"
    Response Write "Mobile=" & mobile & "<br>"

```

```

Response.Write "Fax=" & fax & "<br>"
Response.Write "d_type=" & d_type & "<br>"
if conn.Errors.Count > 0 then
Response.Write(" An Error occured")

Response.Write(conn.Errors(0).Description)
conn.Errors.clear
else
rsmember.AddNew
rsmember.Fields("d_code") = d_code
rsmember.Fields("Name") = name
rsmember.Fields("Specialisation") =
specialisation
rsmember.Fields("Address") = address
rsmember.Fields("Phone") = phone
rsmember.Fields("Mobile") = mobile
rsmember.Fields("Fax") = fax
rsmember.Fields("Sex") = left(sex,1)
rsmember.Fields("EMail") = email
rsmember.Fields("d_type") = d_type
rsmember.update
Response.Write "<h3><font color = 'Green'>"
& "Record Updated!" & "</font></h3>"
end if
end if

```

```

<HR>
<form action="doc.asp" method="post">
<%
if errMsg <> vbNullString then
    Response.Write "<p><font color = 'blue'>" &
    errMsg & "</font></p>"
end if
%>

```

```
<CENTER><FONT >
```

```
<H2><U> Doctor Details</U></H2></FONT></CENTER>
```

```
<CENTER>
```

```
<TABLE cellSpacing=3 cellPadding=3 border=1>
```

```
<TR>
```

```
<TD><FONT color=white size=3>Doctor code</FONT></TD>
```

```
<TD><INPUT name=d_code value="<%=d_code%>"></TD></TR>
```

```
<TR>
```

```
<TD><FONT color=white size=3>Doctor Name</FONT></TD>
```

```
<TD><INPUT name=name value="<%=name%>"></TD></TR>
```

```
<TR>
```

```
<TD><FONT color=white size=3>Specialisation</FONT></TD>
```

```
<TD><INPUT name=sp value="<%=specialisation%>"></TD></TR>
```

```
<tr>
```

```
<td><font color=white>Sex</td>
```

```
<td colspan=2>
```

```
<font color=white> <input type="radio" name="sex" value="Male"
```

```
<% if sex = "Male" then Response.Write "checked" %>> Male </input>
```

```
<font color=white> <input type="radio" name="sex" value="Female"
```

```
<% if sex = "Female" then Response Write "checked" %>> Female
```



```
<A href="fir.asp" ><FONT size=2>Go To Reception</FONT></a>
```

```
<CENTER>
```

```
<TABLE><INPUT id=submit1 type=submit value=submit  
name=submit1> <INPUT id=reset1 type=reset value=" clear "  
name=reset1>&nbsp;
```

```
</TABLE></CENTER></LI>
```

```
</BODY>
```

```
</form>
```

```
</HTML>
```

C. Table design

The table design for all the tables is given below

Table name: Patinfo

Sno	Field Name	Data Type	Constraint
1	Patient-id	Varchar (10)	Primary key
2	Name	Char (20)	
3	Sex	Char (1)	
4	Age	Integer	
5	Address	Varchar (15)	
6	Phone	Varchar (15)	
7	mobile	Varchar (15)	
8	mail_id	Varchar (15)	
9	date_join	datetime	
10	fax	Varchar (15)	
11	B_group	Varchar(5)	

Table name: Inpat detail

Sno	Field Name	Data Type	Constraint
1	Patient-id	Varchar (10)	Primary key
2	D_code	Varchar (10)	Foreign key
3	Desease	char	
4	Test	Varchar (50)	
5	Result	Varchar (50)	
6	Medicines	Varchar (50)	
7	Dc_code	Varchar (10)	
8	Room_type	Varchar (10)	
9	Bed_no	Integer	

Table name: output

Sno	Field Name	Data Type	Constraint
1	Patient-id	Varchar (10)	Primary key
2	D_code	Varchar (10)	Foreign key
3	Desease	char	
4	Test	Varchar (50)	
5	Result	Varchar (50)	
6	Medicines	Varchar (50)	
7	Nfu	integer	
8	Ffu	integer	
9	Last_visit	datetime	
10	amount	money	

Table name: doc

Sno	Field Name	Data Type	Constraint
1	D_code	Varchar (10)	Primary key
2	Name	Char (25)	
3	Spec	Varchar (20)	
4	Address	Varchar (50)	
5	Phone	Varchar (10)	
6	Mobile	Varchar (15)	
7	Mail_id	Varchar (20)	
8	D_type	Char (20)	

Table name: outdoc

Sno	Field Name	Data Type	Constraint
1	D_code	Varchar (10)	Primary key
2	Name	Char (25)	
3	Emp_of	Char(30)	
4	Address	Varchar (50)	
5	V_days	Varchar (25)	
6	V_time	Varchar (25)	
7	Contact_no	Varchar (15)	

Table name: lab

Sno	Field Name	Data Type	Constraint
1	P_id	Varchar (10)	Primary key
2	Lab_id	Varchar (10)	
3	Lab_incharge	Char (25)	
4	T_type	Varchar (20)	
5	Cost	money	

Table name: oper

Sno	Field Name	Data Type	Constraint
1	P_id	Varchar (10)	Foreign key
2	T_id	Varchar (10)	Primary key
3	T_incharge	Char (20)	
4	Type_op	Varchar (30)	
5	Date	datetime	
6	D_incharge	Char (20)	
7	Cost	money	

Table name: password

Sno	Field Name	Data Type	Constraint
1	User_id	Varchar (20)	Primary key
2	password	Varchar (10)	

Table name: playlist

Sno	Field Name	Data Type	Constraint
1	D_code	Varchar (10)	
2	P_id	Varchar (10)	Primary key
3	Doe	datetime	
4	Dod	datetime	
5	L_cost	money	
6	Op_cost	money	
7	R_rent	money	
8	Mis	money	
9	Total	money	

Table name: bed

Sno	Field Name	Data Type	Constraint
1	P_id	Varchar (10)	Primary key
2	Doe	datetime	
3	Dod	datetime	
4	Room_type	Varchar (10)	
5	Bed_no	integer	
6	Avail_room	integer	