

TITLE OF THE PROJECT

HOSPITAL MANAGEMENT SYSTEM

PROJECT PROPOSAL

INTRODUCTION OF THE PROJECT

A Hospital Management System is a computerized software system that manages the function of a health care. The presented project here is made in view overcoming the problems faced by many Hospitals regarding Maintaining patient records, Recording and keeping of Doctor's information, Bed management, Report generation etc. i.e. **"HOSPITAL MANAGEMENT SYSTEM"**.

This project can be widely used in any Hospital which contains different departments with various employees having different designations etc. HMS not only provides an opportunity to the hospital to enhance their patient care but also can increase the profitability of the organization.

About the Organization

A Hospital normally contains information about patient. A hospital management system normally contains the following parts:-

Patient Admission

Patient admission means capturing information of all the patients visiting the hospital and provides record of all the registered patients through a unique patient identification number or room number.

Bed Management

Bed Management means allows hospital authorities to view and track the bed occupancy related operations. Bed Management governs activities like Bed Enquiry, Bed Census, and Bed Maintenance.

Patient Record

The patient record is a comprehensive patient care module that tracks the episodes of care for all patients and maintains and arranges the

clinical information including their address. It is the central repository for holding the entire medical records of the patient. It stores data in an organized and structured manner so that a clinician or any other clinical user easily accesses clinical data based on his role on needs basis.

Our Vision:

We shall define ourselves in the cutting edge technology in the coming era. We shall create honest working environment with see-through-glass planning.

Our Mission:

To create opportunity for growth & self actualization and provide an environment of highly conducive works culture.

OBJECTIVES OF THE PROJECT

The manual system is very slow and not feasible for such type of System.

I have decided to make online the entire management of the patient within the Hospital. In manual system more paper work and involvement of staff is required. So, it is necessary to automate the existing manual system into computerized system.

The main objectives are as follows:

- Improved processing speed in the new computerized system.
- Enhanced capacity to produce a greater amount of activity.
- Providing ability in conducting complex search of records.
- Searching customers/agency detail become easy.
- Cost effective in comparison of manual method.
- Faster retrieval of information.
- Greater accuracy and consistency.
- Reducing paper work.
- Reducing cost, using computing capability to procure data at lower cost along with maintaining accuracy and performance levels.
- Improved arrangement for employee, payments, departments etc.
- Reduces improvement for manpower.
- Less time required for searching data of any department.

PROJECT CATEGORY

This is a RDBMS project. The project belongs to J2EE technology. The language selected is Java, Java Server Pages.

In this project JAVA is the front end which provides an interface for all kinds of works which takes place in the proposed system. MYSQL is the backend for this project. I have used RDBMS based features (like primary and foreign key based in this project) which are provided by any RDBMS packages to keep the data items (stored in tables) of the database intact so that any sort of information doesn't gets lost any kind of query can be performed on it to get the desired results.

TOOLS/ PLATFORM, HARDWARE and SOFTWARE REQUIREMENTS

Hardware Specification

Microprocessor	:	I5 Processor
Ram	:	4 GB (Minimum)
Hard Disk	:	1 TB
Keyboard	:	Samsung 108 keys
Monitor	:	22" Color LG
Mouse	:	Optical Mouse
Printer	:	HP Laser

Software Specification

Operating System	:	Microsoft Windows 10 Professional
Language	:	Java 17 and onwards
Internet Technology	:	JSP
Web Server	:	Tomcat 9.1
Documentation	:	Microsoft Word 2021
Database	:	MYSQL 8.0
IDE	:	Eclipse
Browser	:	Mozilla, Netscape or Internet Explorer
Scripting Language	:	Java Script
Formatting Language	:	Html
Framework	:	Bootstrap

PROBLEM DEFINATION

It takes considerable skill to determine the true cause of the system Problem of the Hospital Management System in Muzaffarpur. Different types of problem; we can find the problem by examining its characteristics.

1. The problem of reliability:

The present system is suffering from the problem of reliability due to its poor management & massive growth in customer.

2. The problem of economy:

The problem of economy of cost associated with billing & employing such a computerized system up to 70% can papers for transactions.

3. The problem of timeless:

The problem of timeless relates more to transmission of information than to the processing & storing on it. The system is suffering from the problem of timeless because the information is available but can't be retrieved when & where it is required.

4. The problem of capacity:

Capacity problems are especially common in the organization that experience peak period of hazard. So there is a need of modification in the current working environment & the computerization is the right option to overcome this problem.

5. The problem of throughput:

The problem of throughput may be viewed as the reverse problem of capacity. Throughput deals with the efficiency of the system if system capacity is high & requirement low, the problem of throughput occurs. Therefore, the problem of throughput is concerned the Hospital Management System.

REQUIREMENT SPECIFICATION

Specification is the culmination of the analysis task. SRS is developed as the consequence of analysis task. Review is essential to ensure that the developer and the customer have the some perception of the system. SRS contains all the things that have been analyzed during analysis task. The introduction of the software requirement specifications provides the goals and objective of the software describing in the context of computer based system. The SRS is nothing more than the software scope of the planning document. The first activity in the software project planning is the determination of software scope. A statement of software scope must be bounded.

REQUIREMENT SPECIFICATION

This phase includes the following:-

- Structural Requirements
- Functional Requirements
- Behavioral Requirements

STRUCTURAL REQUIREMENTS

The proposed computerized system has the following structural requirements:-

- It must have separate database for various information of members like personal details, contact information, etc.
- According to sending and receiving of messages the inbox updating should take place and for this a separate database for inbox updating and maintenance is needed.
- A separate database for keeping the records about the member's status is also needed.
- A database is needed to keep the records about the members associated with each member.
- A database is needed to keep the records about the photos of members.
- A database is needed to keep the records about the videos and links associated with the members.

FUNCTIONAL REQUIREMENTS

The proposed computerized system has the following functional requirements:-

- It requires function for creating user account.
- It requires functions for showing user page.

- A function is needed to make change into the personal and contact details of the member.
- Functions are needed for adding/updating member's status.
- Functions are needed to add photos and videos whenever needed.
- Functions are needed to add and modify the member's details.
- Functions are needed to post comments on status, photos and videos updated by the member himself/herself and by the other members belonging to his/her friend list.
- It requires functions to verify the authentic user for security purpose.

BEHAVIORAL REQUIREMENTS

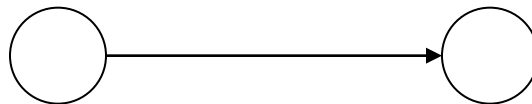
Since this project is not a real time project, hence considerations of behavioral requirement are not so much significant.

PLANNING & SCHEDULING

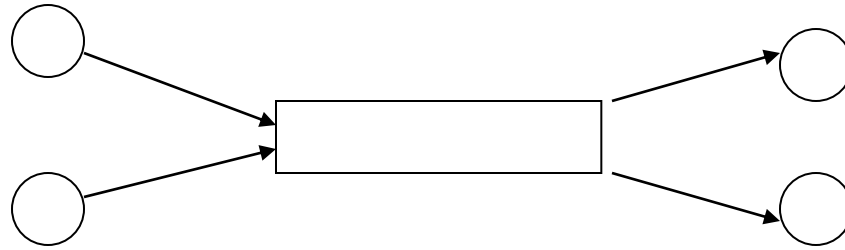
PERT CHART

PERT stands for “program evaluation and review technique” it is used extensively for project planning and control. As a result, the most powerful project planning and management tools are now available to people oriented business. For more than the technical faults, it was found that PERT provided a focus around which managers could brainstorm and put their ideas together. It proves to be great communication medium by which thinkers and planners at level could communicate their ideas, their doubts and fears to another level. It is a diagram in which all activities are shown as a network of dependent activities. It shown:

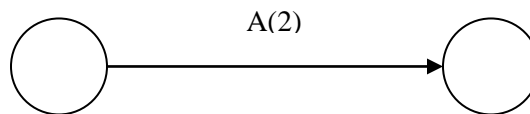
1. Activities: it is represented as an arrow. The tail of an arrow indicates the start of an activity and its head represents the
2. Completion. The start and end events of the activities are shown as circle.



3. Milestones: Milestone marks the end of significant phase. Several activities can start or end at milestones. It is represented as rectangle.



Duration of each task: we show duration of each task by labeling between start and end event of each activity. It is shown that how much time has spent in that activity.



The most important aspect of a PERT diagram is the critical path. A critical path is a chain of critical task. The critical path comprises of critical task, which cannot take longer than estimated without delaying the entire project. The path of activities with zero floats

From the start of the project to the end of the project is called the critical path. A critical path is the longest path on the network.

We passed through such system activities in period of project completion. So, here we identified them as activities and time spend for it. Such system activities are:

.

HOSPITAL MANAGEMENT SYSTEM

System activities:	Time (in days)
A. Project definition	2
B. System analysis	3
C. Feasibility study	2
D. Software engineering	5
E. Software/ hardware requirements	2
F. System design	10
G. Coding	20
H. Validation check	4
I. Testing	4
J. Cost estimation	3
K. User acceptance	5

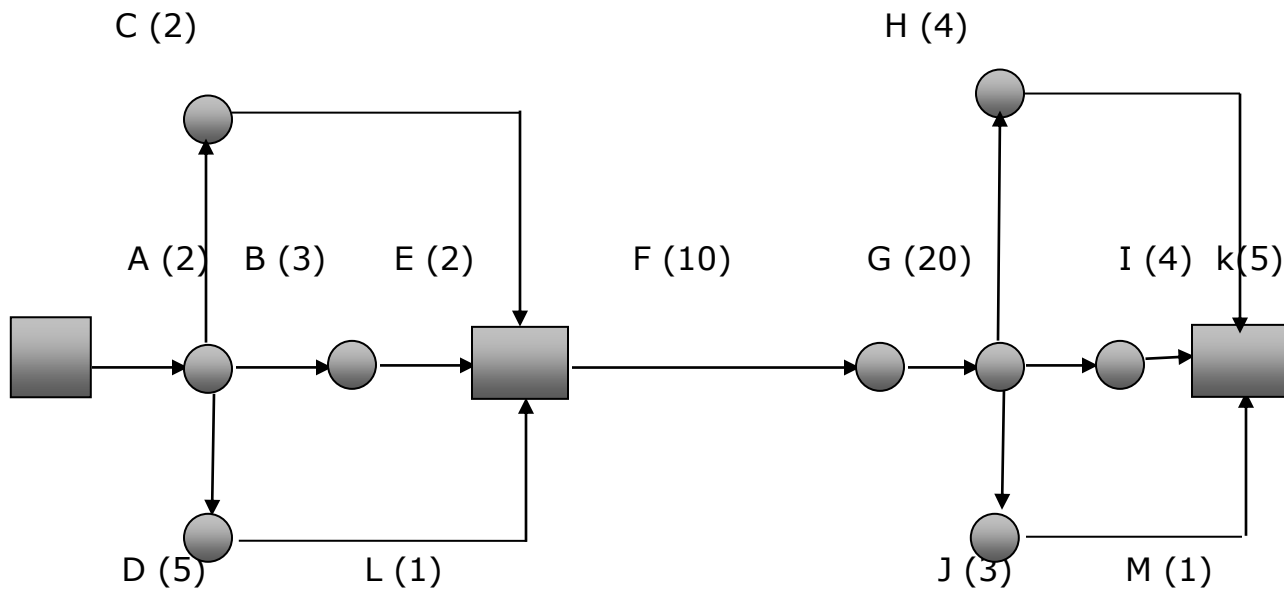
GANTT CHART

It is useful tool for scheduling project activities and tracking is Gantt chart. It is a basic means of presenting a project program visually. A Gantt chart is a bar chart and it gets its name from Harvey Gantt, a pioneer of operation research. Today is largely used to presents the distinct stages of a project. A Gantt chart is composed of vertical axes (usually in day increments) that is, it helps me to plots tasks or resources against time.

We defined vertical axis as system activity and horizontal axis as days. Each system activity is required some time, so, we drawn bar against days for activity. Here also defined time spend in activity. Such as:

<u>System activities:</u>	<u>Time (in days)</u>
A. Project definition	2
B. System Analysis	3
C. Feasibility study	2
D. Software engineering	5
E. Software/Hardware requirements	2
F. System design	10
G. Coding	20
H. Validation check	4
I. Testing	4

HOSPITAL MANAGEMENT SYSTEM



Here , the critical path is, A-> B-> E->F->G->I->K= 46 Days

SCOPE OF SOLUTION

Since this system has been generated by using Object Oriented programming, there are many chances of reusability of the codes in other environment even in different platforms. Also its present features can be enhanced by some simple modification in the codes so as to reuse it in the changing scenario.

The site is made in all possible way to meet the user requirements using latest version of available software and hardware. But as user requirements and operating environment keep changing further extensions can be made on this. In future some more schemas can be added in the "**HOSPITAL MANAGEMENT SYSTEM**" hence these schemas are to be included in the software developed.

ANALYSIS

Requirement analysis is usually the first phase of large-scale software development project. It is under taken after feasibility study has been performed to define the costs and benefits of a software system. The purpose of this phase is to identify and document the exact requirement for the system. In case where the requirements are not clear, interaction and concentration is required between the user and developer. Various software methodologies advocate that this phase must also produce user's manuals and system test plans. In this project the requirement analysis is done with the help of the staffs after considering the needs of the user i.e. what is the basic need of the organization regarding customer and what it is requiring for the automation of the organization after going through the entire requirement we have to come to the next of the software development life cycle.

Following are the main tasks , which has been given by the organization :

- It should have the detailed list of the employees, departments, projects etc.
- It should have the record of each employee and the project associated with it.

A separate module should be designed showing the services Offered like interview. Interview schedules generally begin at the top of the organization structure and work down so as to offend anyone. We also decide to take interview from senior, middle and junior management for getting information about different issues.

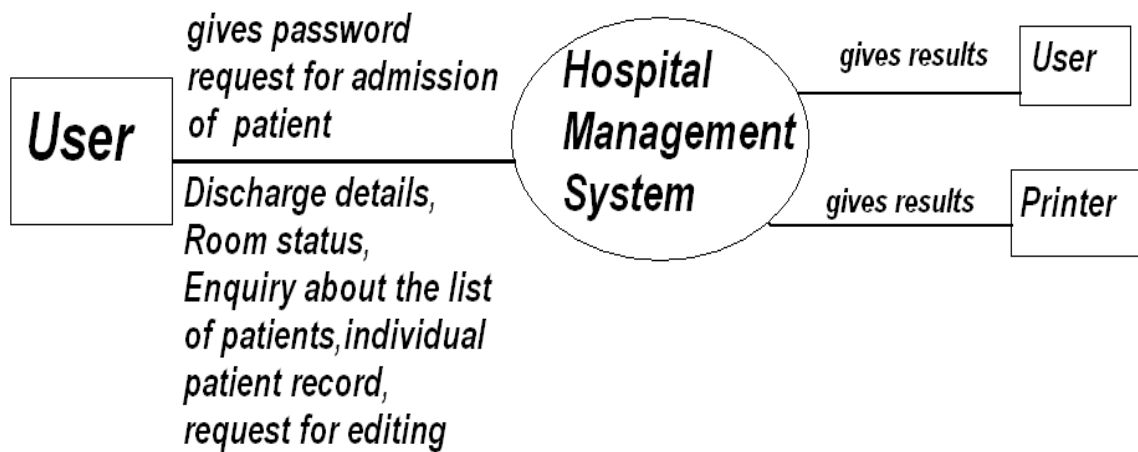
This information, which we got through interview, is very important for designing the system. Question asked from the Staff of the Hospital are as follows:

1. What types of problems they are facing?
2. Up to which extends the records of the organization are secure?
3. How much time is required for verifying the availability of particular employee details?
4. How much accurate the system is?
5. How much time the system will take for seeing detailed list of employee, department etc registration or recruitment Process that have been understand?
6. What is the present working capacity of the Hospital?

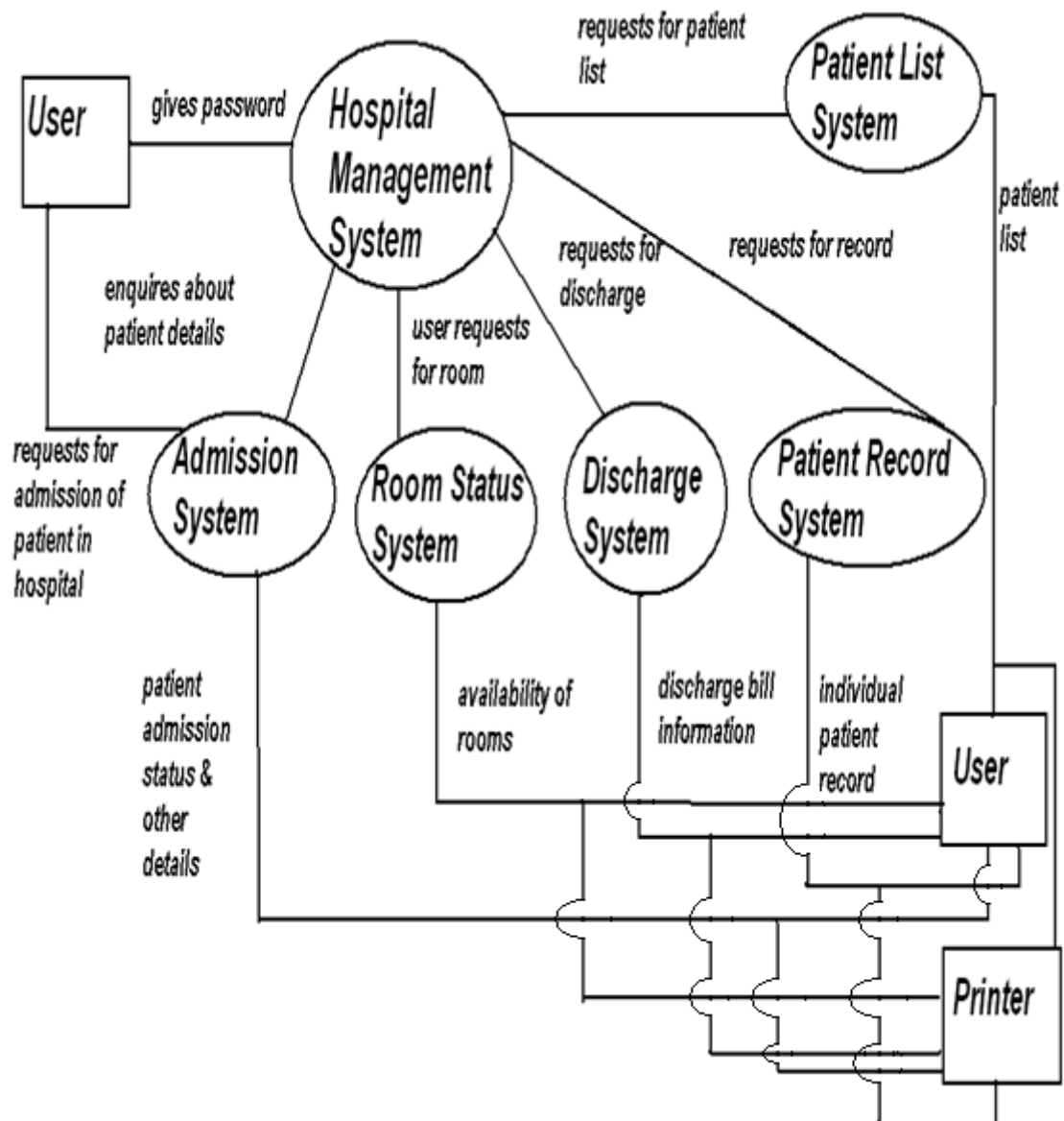
Heavy reliance is placed on the interviewee's report for information about the organization. During this following type of questions are asked.

- ✓ What is being done?
- ✓ How is it being done?
- ✓ What is the process logic?

DFD /ERD

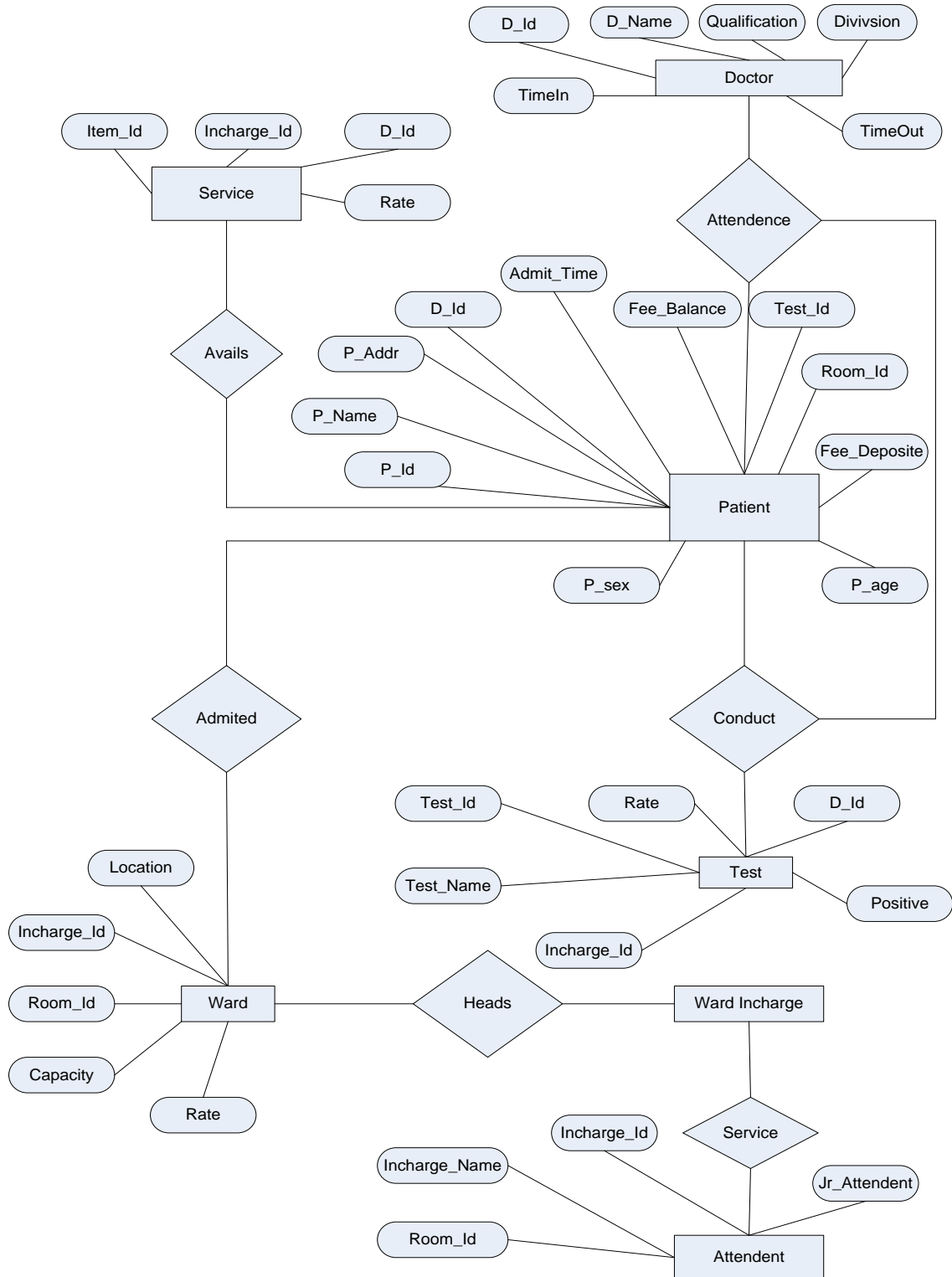


ZERO – LEVEL DATA FLOW DIAGRAM



1st – LEVEL DATA FLOW DIAGRAM

HOSPITAL MANAGEMENT SYSTEM



ER- DIAGRAM

COMPLETE STRUCTURE

MODULES & THEIR DESCRIPTION

The task of the different modules depending upon the nature of work performed by it is as follows:

Login Module

a) Create user module

To login as customer it is mandatory to create a customer account.

In the login module the system will ask following information to be stored in the table.

Customer login

This module will ask username and password and will check it from the oracle Database. If the username and password are correct then it will login. Otherwise it Will display the error message. After login, the customer can book a product and view his last transactions. He can also edit his profile.

Administrator login:

This module will ask username and password and will check it from the oracle Database and will check its privilege. If the username and password are correct and of the administrator then it will login. Otherwise it will display the error message. Further it will redirect to the administrator page from where he can enter the product details, modify the price, stock etc.

Room:

Allow adding details of room, doctor and also allow deleting doctor record.

Admission:

Allow entering details of the patient.

Discharge:

Allow discharging in the patient according to room number and bed number.

Patient list:

Allow displaying the list of the patients such as patient name, room number and bed number.

Patient record:

Search the record of the patient on the basis of room number and bed number.

Edit:

Allow the facility to add rooms, doctor records and to delete doctor record,

Quit:

Allow to exit from the entire software or system.

DATA STRUCTURES FOR ALL THE MODULES:

Login Table

Field Name	Type	Width	Constraints	Description
Uname	String	15	Primary Key	User Name
Upass	String	20	Not Null	Password

SignUp Table

Field Name	Type	Width	Constraints	Description
Uname	String	15	Primary Key	User Name
Upass	String	20	Not Null	Password
Name	String	20	Not Null	Person Name
sex	String	20	Not Null	Gender
Dob	Date	-	Not Null	Date of birth
Country	String	20	Not Null	Country
State	String	15	Not Null	State

HOSPITAL MANAGEMENT SYSTEM

City	String	20	Not Null	City
Street	String	20	Not Null	Street
Zip code	String	10	Not Null	Zip Code
MNo	String	12	Not Null	Mobile Number
e-mail	String	30	Not Null	Email Address
Status	String	15	Not Null	(active/suspended)
Type	String	15	Not Null	(user/administrator)

Hospital:

This table gives static information about hospital.

Field Name	Type	Width	Constraints	Description
HName	String	25	Not Null	Hospital Name
Location	String	30	Not Null	Hospital Address
Hid	String	15	Primary Key	Hospital Id Number

Doctor:

This table contains the information about hospital doctor.

Field Name	Type	Width	Constraints	Description
Did	String	25	Primary Key	Doctor Id Number
DDept	String	30	Not Null	Doctor Department
DName	String	15	Not Null	Doctor Name

Room:

This table contains the information about rooms allocated for patients.

Field Name	Type	Width	Constraints	Description
Roomno	String	25	Primary Key	Room Number
Reg_no	String	30	Not Null	Patient Registration Number
Bed_no	String	15	Not Null	Bed Number
Roomstatus	String	10	Not Null	Availability

Patient:

This table contains the information about patient.

Field Name	Type	Width	Constraints	Description
Reg_no	String	25	Primary Key	Patient Registration

				Number
Mrtstatus	String	20	Not Null	Marital Status
Sex	String	15	Not Null	Gender
P_address	String	15	Not Null	Patient Address
PName	String	20	Not Null	Patient Name

PROCESS LOGIC OF THE MODULES

The Process logic for the various Modules are as:

Main menu:

- Admission
- Discharge
- Room status
- Patient list
- Patient record
- Edit
- Quit

Admission module:

Begin

Enter the name of the patient.

Enter address of the patient.

Enter phone number of the patient.

Enter sex of the patient.

Enter disease of the patient.

Enter serial no. of doctor for the patient.

If want to add more patient then

Go to Begin

Else

Return to the main menu.

End If

End

Discharge module:

Begin

Enter the room number.

Enter the bed number.
If valid room no. and bed no. then
Discharge the patient.
Return the main menu.
Else
Display the error message.
Return to the main menu.
End If
End

Room status module:

Begin
Display the room numbers.
Display the bed numbers.
Display the status (available or not)
Display total numbers of room.
Display total number of occupied bed.
Display total number of available beds.
Return the main menu.
End

Patient list module:

Begin
Display the room number.
Display the bed number.
Display the name of the patient assigned to that
Room number and bed number.
End

Patient record module:

Begin
Enter the room number.
Enter the bed number.
If status is available Then
Display the details of the patient.
Return to the main menu.
Else
Display the error message.
Return to the main menu.
End If
End.

Edit module:

Begin

HOSPITAL MANAGEMENT SYSTEM

Add rooms.
Add doctor records.
Delete doctor record as required.
Return the main menu
End

Quit module:

Begin
Exit from the program.
End.

IMPLEMENTATION METHODOLOGY

System implementation is the stage when the user has thoroughly tested the system and approves all the features provided by the system. The various tests are performed and the system is approved only after all the requirements are met and the user is satisfied.

The new system may be totally new, replacing an existing manual or automated system, or it may be a major modification to an existing system. In either case, proper implementation is essential to provide a reliable system to meet organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system (that is a design question), but improper will prevent it.

Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two persons. Sometimes, they will run both old and new system in parallel way to compare the results. In still other situations, system developers stop using the old system one day and start using the new one the next.

The implementation of the web based or LAN based networked project has some extra steps at the time of implementation. We need to configure the system according the requirement of the software.

For the project we need to install and configure Web logic server 8.1, database server, and the deployment directory for the project.

TESTING PROCESS

In this project, we will start testing from testing the individual module and note all the errors and again debug it. After testing all the individual module we will perform the integration testing in which all the modules are combined together to test it. We note all the error that have arises during the testing and then debug it. Once source has been generated software must be tested to uncover and correct as many errors as possible before delivery to the customer. The test plan is an important product of software design. A test can pre describe various kinds of activities that will be performed to demonstrate that the software product it's required. Any engineered test can be tested on of the following ways: white box testing and black box testing.

The success of testing in revealing errors in programs depends critically on the test cases. In software system the use of testing is not limited to the testing phase. The result of testing is used later on during maintenance also.

These different levels of testing attempt to detect different types of faults. The relation of faults introduced in different phases and the different levels of testing are shown below:

User Needs	Acceptance Testing
Requirements	System Testing
Design	Integration Testing
Code	Unit Testing

REPORT GENERATION

Following are the major categories of Output/Report generation for the proposal of S/W project.

1. Soft and hard copy of the main menu form.
2. Soft and hard copy of Employee Personnel Details.
3. Soft and hard copy of Employee Professional Details.
4. Soft and hard copy of Patient Details.
5. Soft and hard copy of Pay Details.
6. Soft and hard copy of Admission Details
7. Soft and hard copy of Discharge Details
8. Soft and hard copy of date of Joining Details

Verification and Validation

1. It checks the validation of username and password.
2. It verifies the employee code that will work on the terminal.
3. It checks the validation of Employee Id Number.
4. Its security is validated by the password.

IMPLEMENTATION OF SECURITY MECHANISM

Security measures are one of the best efforts to build any information system well secure. The main aim of security measures is to verify that protection mechanism built into a system to protect it from improper penetration.

Security encompasses a set of measures taken to guard against theft, attack, crime and espionage or sabotage. Security implies the quality or state of being secure that is a relief from exposure to danger and acting so as to make safe against adverse contingencies.

Keeping all these views in the mind I also implement a high level of security measures; the user can only use the software after entering the correct password, the password is case sensitive and does not except null entries, if the user enter wrong password he always prompt for reentering the user name and password, and I also provide a feature to change the user name and password options which is used in case of leakage of password, which if happened may cause to break the security measures.

Database / data Security: -

Database security has been imposed in my project by login process. Here the administrator has been given complete authority to access and handle the database in the manner he wants, whereas the general user has been restricted to some specific works over the database.

Creation of User profiles and access rights: -

Only the administrator has been given to create new user and to define access rights for them.

FUTURE SCOPE OF THE PROJECT

In this age of fast growing technology and implementation of the latest technology there is always a scope of future improvement. Also there can be several ways to achieve the target. The main motto is to implement the new concept in the field of development by using the tips and tricks related to the modern, primitive and advance age styles and that makes the difference. There is nothing to do without future applications. So in this project there are also many scopes for future application. This project can be used for any educational organization. By adding some more features it can be also used by central information system also.

Some future scopes of this project are given below: -

- In future it supports multi-user operating system like UNIX.
- In future this project supports distributed database rather than centralized database.
- For better support of user, this project also supports the following techniques like backup file, fast searching technique or queries etc in future.
- Support broad range of input and output devices.
- The database can act as future reference for the organization and passenger's reference.
- Similar other Colleges or University can also use the system for their own use.
- Employees and staff can be trained in some of the relevant parts so that they apply and understand the software – this makes them more aware of use of IT in day to day official dealings.
- A larger complete user-friendly system can be made in future using this application software.

LIMITATION OF THE PROJECT

It is honestly tried to satisfy all the requirements of the Hospital Management System in Project. But there are some limitations in the Project. They are as follows:

- It doesn't make any type of backup.
- It supports sequential searching technique only.
- Only keyboard and mouse is used as input devices.
- Some terms used in this project are based on assumption.
- The system can't take responsibility of unforeseen errors damages and losses due to calamities.
- Changes made in the database may take some time to update or modify.
- Internal processes are not shown by the system.
- System is not platform independent may require some specific configuration to work on the system. The system is windows compatible, may not run on other systems.

BIBLIOGRAPHY

1. System Analysis and Design; Elias M. Awad
2. Fundamental of DATABASE SYSTEM
Ramez Elmas ri, Shamkant B. Navathe Pearson Education
3. Software Engineering A Practioner approach
Roser S. Pressmen PHI
4. Computer Networks
Andrew S. Tanenbaum PHI
5. Object-Oriented modeling and design
James Rumbaugh..
6. An introduction to database management system
Bipin C. desai Galgotia Publication
7. MYSQL The complete Reference
Kevin Loney, George Koch Oracle Press
8. Introduction to JSP The complete Reference
TMH Publication

PROJECT REPORT

INTRODUCTION OF THE PROJECT

A hospital Management System is an automated software system that manages the function of a health care. The presented project here is made in view overcoming the problems faced by many Hospitals regarding Maintaining patient records, Recording and keeping of Doctor's information, Bed management, Report generation etc. i.e. **"General HMS Automation"**.

This project can be widely used in any Hospital which contains different departments with various employees having different designations etc. HMS not only provides an opportunity to the hospital to enhance their patient care but also can increase the profitability of the organization.

About the Organization

A Hospital normally contains information about patient. A hospital management system normally contains the following parts:-

Patient Admission

Patient admission means capturing information of all the patients visiting the hospital and provides record of all the registered patients through a unique patient identification number or room number.

Bed Management

Bed Management means allows hospital authorities to view and track the bed occupancy related operations. Bed Management governs activities like Bed Enquiry, Bed Census, and Bed Maintenance.

Patient Record

The patient record is a comprehensive patient care module that tracks the episodes of care for all patients and maintains and arranges the

clinical information including their address. It is the central repository for holding the entire medical records of the patient. It stores data in an organized and structured manner so that a clinician or any other clinical user easily accesses clinical data based on his role on needs basis.

Our Vision:

We shall define ourselves in the cutting edge technology in the coming era. We shall create honest working environment with see-through-glass planning.

Our Mission:

To create opportunity for growth & self-actualization and provide an environment of highly conducive works culture.

OBJECTIVES OF THE PROJECT

The manual system is very slow and not feasible for such type of System.

I have decided to make online the entire management of the patient within the Hospital. In manual system more paper work and involvement of staff is required. So, it is necessary to automate the existing manual system into computerized system.

The main objectives are as follows:

- Improved processing speed in the new computerized system.
- Enhanced capacity to produce a greater amount of activity.
- Providing ability in conducting complex search of records.
- Searching customers/agency detail become easy.
- Cost effective in comparison of manual method.
- Faster retrieval of information.
- Greater accuracy and consistency.
- Reducing paper work.
- Reducing cost, using computing capability to procure data at lower cost along with maintaining accuracy and performance levels.
- Improved arrangement for employee, payments, departments etc.
- Reduces improvement for manpower.
- Less time required for searching data of any department.

BENEFIT OF NEW SYSTEM

After automation by computerized software the new proposed system will have following benefits over manual system-:

- No manual working i.e. fast working as we need not to maintain same data at number of places
- Fast processing of data
- Searching of records is faster, easier and more complex searches can be easily done which can't be done easily in manual system.
- Automatically, various types of reports can be generated and more complex reports.
- Reduce full involvement of man power.
- Typical queries can be done easily.
- Reduce the cost of stationary.
- New system is less error prone.

DRAWBACK OF OLD SYSTEM

Manual working needs to maintain same data at number of places

- Slow processing of data
- Searching of records is slower, and more complex searches can't be done easily in manual system.
- Various types of reports can be generated, manual reports.
- Full involvement of man power.
- Typical queries can't be done easily.
- The cost of stationary is too high.
- System is error prone.

TOOLS/ PLATFORM, HARDWARE and SOFTWARE REQUIREMENTS

Hardware Specification

Microprocessor	:	I5 Processor
Ram	:	4 GB (Minimum)
Hard Disk	:	1 TB
Keyboard	:	Samsung 108 keys
Monitor	:	22" Color LG
Mouse	:	Optical Mouse
Printer	:	HP Laser

Software Specification

Operating System	:	Microsoft Windows 10 Professional
Language	:	Java 17 and onwards
Internet Technology	:	JSP
Web Server	:	Tomcat 9.1
Documentation	:	Microsoft Word 2021
Database	:	MYSQL 8.0
IDE	:	Eclipse
Browser	:	Mozilla, Netscape or Internet Explorer
Scripting Language	:	Java Script
Formatting Language	:	Html
Framework	:	Bootstrap

ANALYSIS

IDENTIFICATION OF NEED

As Hospital System is an important area, so it is required to have a good software to help organization is managing the day-to-day work and giving right information at right time to help managers in decision making. Very careful analysis has been done to develop this software to meet their need mentioned in the proposal. CBIS (Computer Based Information System) cope up with their practical problems which they were facing in earlier system. The problem is to develop Software that will increase the effectiveness, efficiency, and controlling timelier, relevant accurate information to the concerned area.

I am very eager to know about Hospital Management System and their maintenance system. So, I have decided to choose this topic for my final year project to enhance my knowledge about Hospital Management System. To maintain the data or record about the growing number of Patient, Doctors, Tests, Wards etc. the paper of the work is very tedious task at same time a greater number of human resources is required to the work and number of registers is needed maintain the record.

To reduce the number of human resources, to reduce time of information change within the organization, and to take right decision in the right time it is advisable computerized the whole system.

PRELIMINARY INVESTIGATION

The first step in the SDLC is the preliminary investigation / problem formulation. One must know what the problem is before it can be solved. This phase starts as someone, either a user or a member of particular department. Recognize a problem initiates a request to modify the current computerized system or to computerize the current manual system. The purpose of the preliminary investigation is to evaluate the project requests. When the request is made the first system activity, which is the preliminary investigation begins. An important outcome of the preliminary investigation is determining whether the system request is feasible or not.

The major purpose of this study

1. Identify the responsible users and develop an initial Scope of the system.
2. Identify current deficiencies in the user's environment.
3. Determine objectives for the new system.
4. Determine whether it is feasible to automate the system and if show suggest some acceptable options.

Without minutely studying a system it is not possible to design a new system or change something over and existing system. The first step in my project development is the preliminary investigation to determine the feasibility of the system.

On assigning the preliminary investigation my goal was to first clarify and understand the project request. After that the size of cost and benefits are determined. The technical and operational feasibility determinations are also the important objective from my point of view.

FACT FINDINGS

The data that I collect during preliminary investigations are gathered through three primary methods.

- ☛ Study of the existing system
- ☛ On site observation
- ☛ Reviewing organizational document

All the method mentioned above helped me a lot to know about the existing system, working culture, and many more things. The activity of the organization, office environment, methods of work and facilities provided by service station was minutely observed. But written documents and the on-site observation technique help me to understand how the proposed project should operate.

After going through the record maintained by the schools, and observing the working of the system I gathered a lot of details and information. On the basis of gathered information, I was not able to clearly define the problem, which tensed me a lot. I was totally confused. They do not include enough detailed to allow decision to be made about the merits of a

system proposal, nor do they present user views about current operations.

So I decided to interview the staff of the school. This process of System Analysis helped me a lot in designing the new system.

On the basis of the preliminary investigation, details of the project are given below:

During the preliminary investigation, I consult to the following person and collect the required information:-

-  Doctors
-  Patients
-  Wards
-  Tests
-  Treatment
-  Staff member

This module will provide an easy and simple method updating information related to hospital automation system.

Maintain separate tables that store data related to:-

- Various doctors
- Patients
- Wards
- Treatment
- Staff involving in hospital and its related activities.
- New patient admitted

PROBLEM RECOGNITION

The main problem of the so-called system is how to connect all data base.

The second major problem was all the branches of office are located at different places. It's very difficult to send daily information updating at main site.

- **Evaluation and synthesis:** - Our research indicates that there are mainly three points.

1. Normal requirement: - Type of graphical display, specified system function and defined levels of performance.
2. Expected requirement: - Human machine interaction, overall operational correctness and readability, and ease of software installation.
3. Existing requirement: - we also need such type of available software which is already exist with such system.

Its standard feature is as: -

- **Modeling:** -In system analysis modeling I consider
 - a) What is the basic requirement of customer?
 - b) To establish a basic for the creation of software design.
 - c) To identify a set of requirements that can be validate once the software is built.

To accomplish this objective, the analysis model derived during structure analysis.

- **Specification:** - The software requirement specification deals with

1. Introduction to the software
2. The Information description
3. The Functional description
4. Behavioral description
5. Validation criteria

- **Review:** -

1. Review the product
2. Set an agenda
3. Limit debate
4. Write notes
5. Develop the checklist
6. Allocate the resources
7. Review all early review

FEASIBILITY STUDY

An important outcome of the preliminary investigation is the determination whether the system requested is feasible or not. That requires the need for a rigorous feasibility study.

Once scope has been identified, it is reasonable to ask: can we build software to meet this scope? Is the project feasible? Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible, I can go ahead and prepare the project specification which finalizes project requirement.

Generally, feasibility studies are undertaken within tight time constraints and normally culminates in a written and oral feasibility report. The contents and recommendations of such a study will be used as a sound basis for deciding whether to proceed, postpone or cancel the project.

In the conduct of the feasibility study, I considered only four types of feasibility

- Economic feasibility
- Technical feasibility
- Operational feasibility
- Time feasibility

◆ **Economical Feasibility:**

It is very most important. Because software is the most expensive element for virtually all computer-based system. A large cost estimation error can make the difference between profit and loss. cost overrun can be desirous for the developer. Here I used economic analysis to study economic feasibility. Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as **cost/benefit** analysis: the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with cost. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have chance of begin approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

A system that can be developed technically and that will be used if installed must still be profitable for the organization. Financial benefit must equal or exceed the costs. The analysts arise various financial and economic questions that during the primary investigation to estimate the following:

- The cost to conduct a full systems investigation.
- The cost of hardware and software for the class of application being considered.
- The benefit in the form of reduced costs or fewer costly errors

- The cost if nothing changes

◆ **Cost/Benefit Analysis:**

To carry out an economic feasibility study, it is necessary to place actual money values against any purchases or activities needs to implement this project. It is also necessary to place money values against any benefits that will accrue from a new system caused by the place money values against any benefits that will accrue from a new system created by the project. Such calculations are often described as cost-benefit analysis.

◆ **Cost-benefit analysis usually includes two steps:**

1. Producing the estimates of cost and benefits,
2. And determining whether the project is worthwhile once these costs are ascertained.

◆ **Producing Costs and Benefits:**

The goal is to reduce a list of what is required to implement the system and a list of the new system benefits.

Cost-benefit analysis is always clouded by both tangible and intangible items. Tangible items are those to which direct values can be attached (e.g. the purchase of equipment, time spent by people writing programs, or items such costs or the gst). Some tangible costs often associated with computer system development are: -

- **Equipment costs for the new systems:** various items of computing equipment, as well as items such accommodation costs and furniture, are included here
- **Personal costs:** These include personnel needed to develop the new system and those will subsequently run the system when it is established. Analysis, designers and programmers will need to build the system. Also include the personnel to train the system users.
- **Material costs:** These include stationary, manual production and other documentation costs.
- **Conversion costs:** The costs of designing new forms and producers and of the possible parallel running of the existing and new systems are included here.
- **Training costs:** These include the cost of training users of the new systems, as well as developers who may be required to use new technologies.
- **Other costs:** Sometimes consultant's costs are included here together with the management overheads, secretarial support, travel budgets, and so on.

Intangible items, on the other hand, are those whose values cannot be precisely determined and are the result of subjective judgment. For example, how much is saved by completing a project earlier or providing

new information to decision makers? Considerable argument can take place before agreement is reached on such intangible costs.

The sum value of costs of items needed to implement the system is known as the system cost. The sum value of the saving made is known as the benefit of the new system. Once we agree on the costs and benefits, we can evaluate whether the project is economically feasible.

The cost estimates are usually used to set the project budget. Often it is convenient to divide these costs into project phases to give management an idea of when funds and personnel will be needed. The cost estimate needs to be worked out very carefully. We will avoid omitting anything from estimates, as this will necessitate requests for more funds because something was forgotten.

On the other side of the evaluation are benefits of the project, which may also tangible or intangible. Tangible benefit includes those benefits that can be measured in actual rupees terms. Such benefit can include reduced production costs through the introduction of new technologies or reduced recessing costs through the use of computers.

Intangible benefits such as more satisfied customer or improved cop rate image because of using new system are not easily quantified.

Both tangible and intangible costs and benefit will be asked consideration in the evaluation process

◆ **Determining Whether This Project is Worthwhile:**

The costs and benefit are used to determine whether this project is economically feasible or not. There are two ways to do this: the payback and the present value method. Here we use the **payback** method.

The payback method: The payback method defines the time required to recover the money spent on the project. The concept is quite simple. We know when this project will start. We also know the costs and benefits for each succeeding year. The difference between the cost and the benefit for each year will be saving or net benefit for the year. The computation is used to determine the number of years. The computation to determine the number of years needs to recover the cost is quite simple.

◆ **Technical Feasibility:**

This concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but I consider the following:

- The facility to reduce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.
- Facility to communicate data to distant location.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. the configuration should give the complete picture about the system's requirement:

There are a number of technical issues which are generally raised during the feasibility stage of the investigation. They are as follows:

- Q.** Does the necessary technology exist to do what is suggested (and can it be acquired)?
- Q.** Does the proposed equipment have the technical system?
- Q.** Can the system be upgraded if developed?
- Q.** Are there technical guarantees of accuracy, reliability, ease of access and data security?
- Q.** Does the necessary technology exist to do what is suggested and can it?
- Q.** How many workstations are required?
- Q.** How these units are interconnected so that they could operate and communicate smoothly?
- Q.** What speeds of inputs and output should be achieved at particular quality of printing

Specific hardware and software products can then be evaluated keeping in viewing with the logical needs.

◆ **Operational Feasibility:**

Proposed projects are beneficial only if they can turn into information

systems that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed. It is mainly related to human organizational and political aspect. Some of the important questions that are useful to test the operational feasibility of a project are given below:

- Is proposed system operational from the management and users' point of view? If the present system is well liked and used to the extent that persons will not be able to see reasons for a change, there may be resistance.
- Are current business methods acceptable to the users? If they are not, users may welcome a change that will bring about a more operational and useful system.
- Have the users been involved in the lining and development of the project? If they are involved at the earliest stage of project development, the changes of resistance can possibly reduce.
- Will the proposed system cause harm? Will it produce poorer result in any case or area? Will the performance of staff member fall down after implementation?
- What organizational structures are distributed?
- What new skills will be required?
- Do the existing staff members have these skills?
- If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

This feasibility study is carried out by a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and skilled in system analysis and design process.

◆ **Time Feasibility:**

Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame. If this project takes too much time it is likely to be rejected.

- I as a system developer will take as a series of tasks, each of which will have a specific scope and completion date.
- Review of the current system, interviews with the people and review of documentation will take approximately one person two weeks to complete and will be performed by me.
- Concerting other programs venue will take approximately one person one week to complete and will be performed.
- Analysis of facts, determination of feasibility, definition of user requirements and direction will take two people approximately two weeks to complete and be performed by both analyst (80 persons hours) after completion of task 1 and 2.

PROJECT PLANNING

After preliminary study, I get sufficient information about the proposed project. On the basis of the information collected, I re-evaluated my project scope. Firstly I shorted out different types of requirement of the proposed system, and then grouped them into different groups on the basis of the importance of requirements. Some of them were totally absurd i.e. of no importance as expected before the beginning investigation. I set the priority of every requirement So that I could plan further steps on the basis of priority set. I have performed almost all the task like assessing risk involved in the project, function & feature of the project. I have also defined the technical functions & features involved in the project planning. They are included in the coming pages of the project report.

The objective of software project planning is to provide a framework that enables us to make, reasonable estimates of resources, cost and schedule. The project plan must be adapted and updated as the project proceeds.

Task sets for project planning.

Establish Project Scope: -

Day by day the Hospital will spread its business and go long in near future. So usability of this software will also increase.

1. Determine Feasibility

Feasibility of this project is studied and as a result of this study, it is found that this project is economically, technically and operationally feasible.

2. Analyze Risks

Risk analysis is a series of steps that help a software team to understand and manage uncertainty. Risk always involves two characteristics

Uncertainty: - A risk is a potential problem it might happen, it might not.

Loss: - If risk happens, losses will occur

There are many types of risks

- Projects risks (threaten the project plan)
- Technical risks (The quality and timeliness of the software to be produced)
- Business risks (Viability of the s/w to be built)

Since the developing project is software for the Examination and it is developed for the purpose of submitting as a final project. Again, its contents always update information. So, no any critical risk is identified.

3. Define Requirement resources: -

- This project requires a very few human resources such as an operator to carry out the work.
- Html is used as front end.
- Java JSP is used for business logic.
- Oracle is required as back end.

5. Estimate Cost & Effort

Here estimating cost is irrelevant. Since this project is being developed by me only for the submission of the final project. So, there is no need to

decompose the problem, develop two or more estimates using size, function points, process tasks, or use-cases.

PROJECT SCHEDULING

This project is a database application. It has relatively fine granularity for tasks to be performed in the short term.

The Project Schedule provides a road map for a s/w Project manager if it has been properly developed. The project schedule defines the tasks and milestones that must be tracked and controlled as the project proceeds.

There are two project scheduling methods PERT and CPM (Program Evaluation and Review Technique & Critical Path Method) that can be applied to s/w development. Project should be organized into modules or segments so that its progress can be easily measured. If there is a problem in one module, we can concentrate our mind over that problem only and we can achieve a optimum solution. A schedule must be flexible because it may create some problem (project delay). So, if it is flexible, coming milestones can be adjusted easily. A schedule has two primary functions it is both a plan and a device for measuring program. The key steps in a schedule are called milestones or checkpoints. As the Project progress, the date of each milestone is compared with the date of predict milestone. When a project is behind Schedule, corrective steps must be taken. There are so many options

HOSPITAL MANAGEMENT SYSTEM

- Increase the budget
- Increase the manpower
- Add Equipment
- Change Priorities

SOFTWARE REQUIREMENT SPECIFICATION

Specification is the culmination of the analysis task. SRS is developed as the consequence of analysis task. Review is essential to ensure that the developer and the customer have some perception of the system. SRS contains all the things that have been analyzed during analysis task. The introduction of the software requirement specifications provides the goals and objective of the software describing in the context of computer-based system. The SRS is nothing more than the software scope of the planning document. The first activity in the software project planning is the determination of software scope. A statement of software scope must be bounded.

In the context of computer-based system (and software), the term specification means different things to different people. A specification can be a written document, a graphical model, a formal mathematical model, a collection of usage scenarios, a prototype, or any combination of these.

Some suggest that a "standard template" (SOM97) should be developed and used for a system specification, arguing that this leads to requirements that are presented in a consistent and therefore more understandable manner. However, it is sometimes necessary to flexible when a specification is to be developed. For large system, a written document, combining natural language description and graphical models may be the

best of approach, however, usages scenarios may be all that are required for smaller products or systems that reside within well understood technical environments. The systems specification is the final work product produced by the systems and requirements engineering, database engineering, and human engineer. It describes the function and performance of a computer-based system and the constraints that will govern its development. The system specification also describes the information (data & control) that is input to and output from the system.

Software scope describes the data & control to be processed, function, performance constraints, interfaces & reliability.

Introduction

The detail introduction I have mentioned in the starting of project report.

Objectives

Objective is given in Detail, in Project report in starting.

The information description

It provides a detail description of the problem that the software must solve. I have stated the goals and objectives of the software, describing it in the context of the computer-based system and provided a detailed description of the problem that the software must solve.

Functional description: It provides a detail description of each functions required to solve the problem. Login process, details gathering process, services displaying process, online application process etc. are some functions of the system which solve the problem of the organization. A clear DFD is drawn to show all functions.

Behavioral description: - It examines the operation of the software as consequences of external event & internally generated control characteristics. A clear STD is drawn to show states and events.

These are some of the Behavioral Description present in the software:

- Work well in adverse conditions.
- Work as for the requirements.
- Work with reliability, correctness, and easiness.
- Work without any external support

Validation criteria: - It examines the implicit review of all other requirements. Generally, we neglect this section in SRS, but it plays a vital in SRS. In this we consider what classes of test must be conducted to validate function, performance & constraints. The rest has been declared in the Testing field.

During system study it is found that many other requirements are needed for existing system:

- A System must have Graphic User Interface.
- A should be a system for automatic calculations like calculation of treatment fee, tests result calculation.
- Prohibit illegal operation on to the mark sheet or unauthorized access of records.
- Very few data should be inputted while demand customer records i.e. the basic details, previous dues, installment amount etc must be retrieved automatically.
- System must be accurate, secure, reliable and fast.
- There must be provision for Report Generation as and when required and also provision of printing of the Report.
- Registration number, automatically and should be in serial number.
- There should be standard short cut for the operation in the proposed project.

Specification is the foundation of the development phase. So much more concentrations should be given at the time of SRS. It includes Bibliography and Appendix which are given in project report in their subsequent section.

Problem definition

It takes considerable skill to determine the true cause of the system Problem of the Hospital System in Muzaffarpur. Different types of problem; we can find the problem by examining its characteristics.

1. The problem of reliability:

The present system is suffering from the problem of reliability due to its poor management & massive growth in customer.

2. The problem of economy:

The problem of economy of cost associated with billing & employing such a computerized system up to 70% can papers for transactions.

3. The problem of timeless:

The problem of timeless relates more to transmission of information than to the processing & storing on it. The system is suffering from the problem of timeless because the information is available but can't be retrieved when & where it is required.

4. The problem of capacity:

Capacity problems are especially common in the organization that experience peak period of hazard. So, there is a need of modification in the current working environment & the computerization is the right option to overcome this problem.

5. The problem of throughput:

The problem of throughput may be viewed as the reverse problem of capacity. Throughput deals with the efficiency of the system if system capacity is high & requirement low, the problem of throughput occurs. Therefore, the problem of throughput is concerned the Hospitals.

SOFTWARE ENGINEERING PARADIGN APPLIED

Software system passes through two phases during their life cycle.

1. The development phase
2. The operational maintenance phase

The development phase begins when the need for the product is identified, it ends when the implemented product is tested and delivered for operation. Operation and maintenance include all activities during the operation of software such as fixing bugs discovered during operation, making performance enhancement, adapting the system to its environment, adding minor feature.

Here we follow the waterfall model to develop this project. In this model following steps are performed to develop software: -

Problem definition: -

The first development of this phase process understands the problem in question and its requirements. At this point, the managers and software specialist decide whether it feasible to build to the system.

Analysis: -

Analysis phase delivers requirement specification. specification must be carefully checked for suitability, omission, inconsistencies and ambiguities. In this project analysis is done with the help of Hospital staff and in charge considering the need of computerizing the Hospital i.e. what is the basic need and the requiring for the automation of the Hospital. After going the

entire requirement, we have come to the next phase of the software life cycle.

Design: -

Design process of mapping systems requirement defined during an analysis to an abstract representation of a specific system implementation, meeting the cost and performance constraints. The purpose of design is to specify a particular software system. The modules may be further decomposed into sub module and procedures until each module can be implemented easily.

Validation check: -

Software validation is achieved through a series of black box test that demonstrate conformity with requirements. At the plan outlines that classes of test to be conducted and a test procedural define specific test will used that demonstration conformity with requirements. Is my project got its motive. Is it's fulfilling the need of customer and who will used the software and is used satisfied with my work.

- Transportability → The software is transportability i.e may carry from one place to another place without any problem.
- Compatibility → This software is compatibility to any system so it can be use in any environment.
- Error recovery → Error recovery is easy, if any error will occur in this program at this stage User or Administrator can easily handle all the error.

- Maintainability → The important structure is maintainability. this software is well maintained and its user friendly at any stage. So, the proposed site and software is well structure and error can handle friendlier.

Implementation & Maintenance: -Once the system passes all the tests, it is delivered to the customer and enters the maintenance place. Any modification made to the system after initial delivers are usually attributed to the phase. Basically, in this phase the software that I have received the software and assured that all documentation is available.

An implementation description of software provides the internal details of the required procedures. First, we implement a prototype we consider all the system throughout.

6. Corporative maintenance

- 1) Adaptive maintenance
- 2) Perfective maintenance

THE LINEAR SEQUENTIAL WATERFALL MODEL

It is an essential approach to software development that begins at the system level and progresses through analysis, design, coding, testing & support. It encompasses the following activities: -

- System / information engineering

- Software requirements analysis
- Design
- Code generation
- Testing
- Support (Maintenance)

System / Information Engineering & Modeling: -

In it all the system elements are identified and then requirements for all the system elements are established.

Software Requirements Analysis: -

To understand the nature of the program(s) to be built, the software engineer must understand the information domain for the software, as well as required function, behavior, performance and interface. Requirements for both the system and the software are documented and reviewed with the customer.

Design: -

It is actually a multi-step process that focuses on four distinct attributes of a program i.e. data structure, software architecture, interface representations, and algorithmic detail. It is also documented.

Code Generation: -

This step translates the design into a machine-readable form.

Testing: -Once code has been generated, program testing begins. It is conducted to uncover errors and ensure that defined input will produce

actual results that agree with required results.

Support: -

Software support / maintenance reapplies each of the preceding phases to an existing program rather than a new one.

I am presenting here waterfall model to represent the different phases....

Component of Software Engineering

The important component of software engineering paradigm is:

- Software Development Life Cycle- It defines the various stages and activities associated with the development of a software system.
- Software Quality Assurance- This is the process of ensuring user satisfaction through the development of a quality product.
- Software Project Management- It is the application of the principles project management to the process of software development.
- Software Management- They is the methods and procedures to be followed for effective software maintenance and change control.
- CASE (Computer Aided Software Engineering) - They are a set of automated tools that support the process of software development.

Data model: - The most important and difficult task for many databases application in the database design. The design of accompanying programming code is usually much easier. We should design a database for

many of the same reasons that we should design any computer program: careful design of software before coding improves the quality and reduces the cost. A database design is often referred to as a *data model* or schema. In general, there are two approaches to database design. The first approach is attribute driven: compile a list of attributes relevant to the application and synthesize groups of attributes that are meaningful to the application and describe them. In a typical design, there are ten times fewer entries than attributes, so entity design is much more tractable. Object modeling is a form of entity design.

Control flow diagram (CFD): - A data flow diagram shows all possible computation path for values; it does not show which path are executed and in what order. Decision and sequencing are control issues that are part of the dynamic model. A decision affects whether one or more functions are even performed, rather than supplying a value to the functions it is sometimes useful to include them in the functional model so that they are not forgotten and so their data dependencies can be shown.

Including control flow in the data flow diagram does this.

A control flow is a Boolean value that affect whether a process is evaluated. The control flow is not an input value to the input values to the process itself. Control flow can occasionally be useful, but they duplicate information in the dynamic model and should be used sparingly.

State diagram: - A state diagram allows us to further explore the operation attribute that need to be defined for an object. They consist of a set of states that an object is in and event, which take the object from one state to another.

Or in other words a state diagram relates events and states. When an event is received, the next state depends on the current state as well as the event; a change of state caused by an event is called a transitions. A state diagram is a graph whose nodes are states and whose directed arc is transition labeled by event names. A state is drawn as a rounded box containing an optional name. A transition is drawn as an arrow from the receiving state to the target state; the label on the arrow is the name of the event causing the transition. All the transition leaving the state must correspond to different events.

The state diagram specifies the state sequence caused by an event sequence. If an object is in a state and an event labeling one of its transitions occurs, the object enters the state on the target end of the transition. The transition is said to fire. If more than one transition leaves a state, then the first event to occur causes the corresponding transition to fire. If an event occurs that has no transition leaving the current state, then the event is ignored. A sequence of events corresponds to a path through the graph.

A state diagram describes the behaviors of a single class of objects.

Since all instances of a class have the same behaviors, they all share the same state diagram, as they all share the same class features. But as each object has its own attribute values, so too each object has its own state, the result of the unique sequence of events that it has received. Each object is independent of other objects and proceeds at its own pace.

State diagram can represent one-shot life cycles or continuous loops. One-shot diagrams diagram represents objects with finite lives. A one-shot diagram has initial and final states. The initial state is entered on creation of an object; entering the final state implies destruction of the object. The solid circle shows an initial state. The circle can be labeled to indicate different initial condition. A final state is shown by bull's eye. The bull's eye can be labeled to distinguish final conditions. A one-shot diagram can be considered a state diagram "subroutine" that can be referenced from various places in a high-level diagram.

The *dynamic model* is a collection of state diagrams that interact with each other via shared events. An object model represents the static structure of a system, while a dynamic model represents the control structure of the system. A state diagram like an object class is a pattern; it describes an entire, possibly infinite, range of sequences. A scenario is to a dynamic model as an instance diagram is to an object model.

Sequence diagram: - The second type of behavioral representation, called sequence diagram in UML, indicates how events cause transition from

object to object. Once examining a use-case has identified events, the modeler creates a sequence diagram-a representation of how events cause flow from one object to another as a function of time. In essence, the sequence diagram is a shorthand version of the use-case. It represents key classes and the events that cause behavior to flow from class to class.

Once a complete sequence diagram has been developed, all of the events that cause transition between system objects can be collated into a set of input events and output events (from an objects). The information is useful in the creation of an effective design for the system to be built.

Class Diagram: - A class diagram is used to support functional requirement of system. In a static design view, the class diagram is used to model the vocabulary of the system, simple collaboration, and logical schema. It contains set of classes, interfaces, collaboration, dependency, and generalization and association relationship. The notation for classes and relationship between classes as shown as:

Or in other words we can say that a class diagram is a schema, pattern, or template for describing many possible instances of data. A class diagram describes object classes.

An instance diagram describes how a particular set of objects relates to each other. An instance diagram describes object instances. Instance diagram is useful for documenting test case (especially scenarios) and discussing examples. A given class diagram corresponds to an infinite set

of instance diagram.

In given figure below shows a class diagram (left) and one possible instance diagram (right) described by it. Objects A, B & an anonymous person are instances of class person. The OMT symbol for an object instance is a rounded box. The class name in parentheses is at the top of the object box in boldface. Object names are listed in normal font. The OMT symbol for class is a box with class name in boldface.

Class diagram describes the general case in modeling a system. Instance diagram are used mainly to show examples to help to clarify a complex class diagram. The distinction between class diagram and instance diagram is in fact artificial; classes and instances can appear on the same object diagram, but in general it is not useful to mix classes and instances.

CRC (Class-Responsibility-Collaborator Modeling): - Once basic usage scenarios have been developed for the system, it is time to identify candidate classes and indicate their responsibilities and collaboration. Class responsibility- collaborator (CRC) modeling provides a simple means for identifying and organizing the classes that are relevant to system or product requirements. Ambler describes CRC modeling in the following way:

A CRC model is really a collection of standard index card that represent classes. The cards are divided into three sections. Along the top of the card we write the name of the class. In the body of the card, we list the class

responsibilities on the left and the collaborators on the right.

In reality, the CRC model may make use of actual or virtual index cards.

The intent is to develop an organized representation of classes.

Responsibilities are the attributes and operations that are relevant for the class. Stated simply, a responsibility is “anything the class knows or does”. Collaborators are those classes that are required to provide a class with the information needed to complete a responsibility. In general, collaboration implies either a request for information or a request for some action.

Classes

To summarize, objects manifest themselves in a variety of forms: external entities, things, occurrences, or events; roles; organizational units; places; or structures. One technique for identifying these in the context of software problem is to perform a grammatical parse on the processing narrative for the system. All nouns become potential objects. However, not every potential object makes the cut. Six selection characteristics were defined:

- **Retained information:** The potential object will be useful during analysis only if information about it must be remembered so that the system can function.
- **Needed services:** The potential object must have a set of identifiable operations that can change the value of its attributes in some way.

- **Multiple attributes:** During requirements analysis, the focus should be on “major” information an object with single attribute may, in fact, be useful during design but is probably better represented as an attribute of another object during the analysis. Activity.
- **Common attributes:** A set of attributes can be defined for the potential object and these attributes apply to all occurrences of the object.
- **Common operation:** A set of operation can be defined for the potential object and these operations apply to all occurrences of the object.
- **Essential requirements:** External entities that appear in the problem space and produce or consume information that is essential to the operation of any solution for the system will almost always be defined
 - as object in the requirement model.

A potential object should satisfy all six of these selection characteristics if it is to be considered for inclusion in the CRC model. Firesmith extends this taxonomy of class type by suggesting the following addition.

Device Classes model external entities such as sensor, motors, keyboards

Property classes represent some important property of the problem environment (e.g., credit rating within the context of a mortgage loan application).

Interaction classes model interaction that occurs among other objects (e.g. a purchase or a license).

In addition, objects and classes may be categorized by a set of characteristics.

Tangibility. Does the class represent a tangible thing (i.e. a keyboard or sensor) or does it represent more abstract information (i.e. a predicated outcome)?

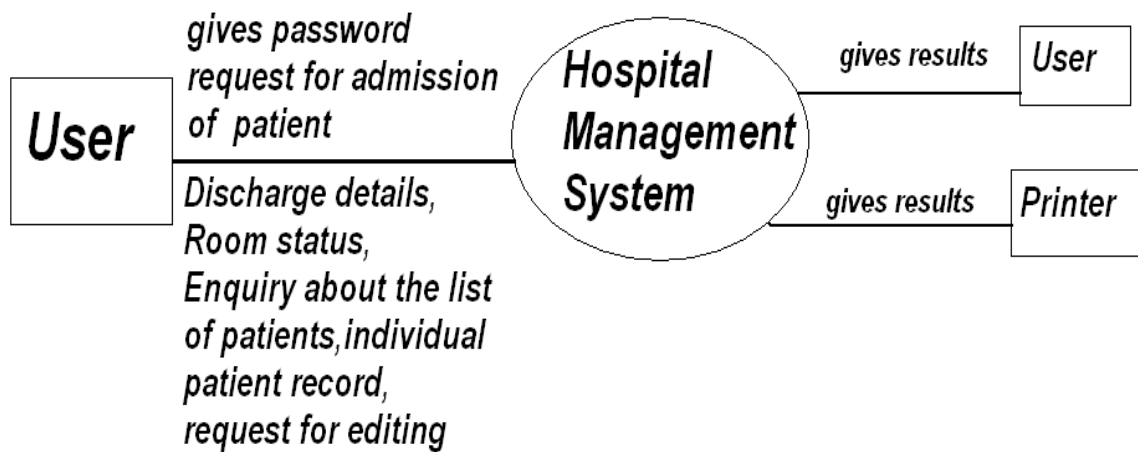
Inclusiveness. Is the class atomic (i.e. it has its own thread of control) or sequential (it is controlled by outside resources)?

Persistence. In the class transient (i.e. it is created and removed during program operation), temporary (it is created during program operation and removed once the program terminates), or permanent (it is stored in a database)?

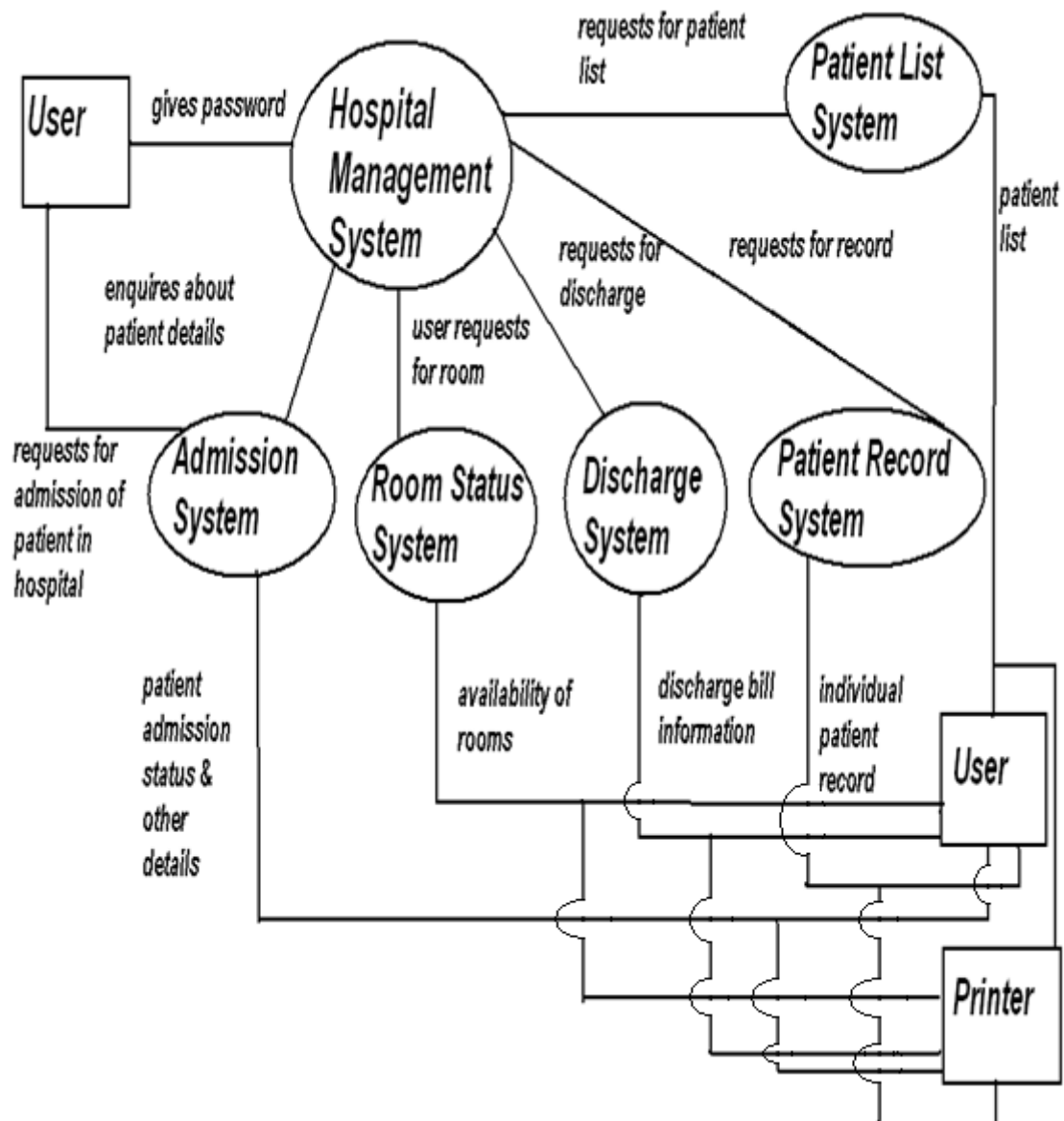
Integrity. Is the class corruptible (i.e. it doesn't protect its resources from outside influence) or guarded (i.e. the class enforces controls on access to its resources)?

DFD/ERD

DFD /ERD



ZERO – LEVEL DATA FLOW DIAGRAM



1st – LEVEL DATA FLOW DIAGRAM

DESIGN

SYSTEM DESIGN

Design is much more creative process than analysis. It is working with the unknown new system, rather than analyzing the known system. It is even harder to define the best design for a given system. A good design is very dependent on a particular system, and what is a good design for one system may be bad for another. System design requires considerable creativity to propose the necessary system-specific changes and additions that are both acceptable to users and easy to implement.

The figure of linear design cycle, which is given below, specifies the stage when design begins formally. It even requires objective to be specified before a designer begins the design. This simplifies the design process by focusing it on particular objectives. Design is directed at searching out ways to meet the objectives. There are many types of objectives. Common types are:

1. **Functional objectives**→ It states a new or amended functional system requirements. For example:
 - New or changed output reports or displays;
 - New services to be performed by the system functions; and
 - Revised security of access controls.
2. **Process improvements**→ It includes:
 - Changes to the ways the data is accessed;
 - Changes of the sequence in which things are done;

- Changes to the process steps; and
- Changes to input and output methods

3. **Operational objectives**→ It specify performance standards to be attained by the new system. These define system accuracy and various timing requirements.

4. **Personal and job satisfaction needs**→ Important objectives include designing systems that are easy to use and which allow users to be creative rather than simply responding to computer outputs.

Different design ideas are needed to satisfy each of these objectives.

Functional objectives may require change to the kind of computations in the system. **Operational objectives**, on the other hand, require different solutions. They may, for example, require a different physical implementation of some of the system processes to improve performance or to improve flexibility. **Personal and job satisfaction objectives** may call for changes to the user interface to the computer, new reports layout or changes to the data flows.

Design must address all these issues. It must specify the layout of any input screens or forms. The outputs are also designed to show the layout of reports or screens.

This phase produced a design for the proposed system. We selected the equipment needs to implement the system. In this stage, we specified new programs or changes to existing programs, as well as file system. We also

produced detailed procedures that describe how user will use the system. In above figure, we see that we developed and passed through different phases to keep the validations and requirements of system users. We categorized **System Design** usually in two steps: **broad design** and **detailed design**.

Broad Design

During this phase, the conceptual solutions proposed by the feasibility study are looked at in more detail. Major new functions are proposed and change to existing functions defined. Important inputs and outputs are also defined at this point and performance requirements are specified. In addition, broad design outlines which part of the system is to be automated and which will remain manual.

At the conclusion of broad design, we knew what we needed in order to build the system. This included the size of the computer and the software needed to put the system together. It identified the main architecture of the proposed system. This architecture is verified against the proposed system model and validated against user requirements.

Detailed Design

It is only when a broad design chosen that detailed design starts. During the detailed design phase, the database and program modules are designed and detailed user procedures are documented. The interaction between the computers and system users is also defined. These interfaces

defined exactly what the user will be expected to do to use the system.

The output of detailed design included a proposed equipment configuration together with specification for the database and computer programs.

Detailed user procedures are also provided. These included any input forms and computer interaction between users and computer. The user manual is also readied at the end of this phase. This output, particularly the proposed interactions between the computer and users, is validated against requirements.

DESIGN OBJECTIVE

The preliminary objective of the design of my project is to deliver the requirement as specified in the feasibility report. However, I kept following design objectives during the designing phase: -

- **Efficiency:** - This involves accuracy, timeliness and comprehensiveness of the system output of the said project.
- **Practicality:** - The system must be stable and can be operated by people with average intelligence; however, I provide the necessary training to the operator to work in properly on my project.
- **Cost:** - It is one of the important considerations. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.
- **Flexibility:** - The system should be modifiable depending on the changing needs of the user or Bank. Such modification should not

entail extensive reconstructive or recreation of software. It should also be portable to different computer system.

- **Security:** - This is very important aspect of the design and should cover the areas of hardware reliability, fall back procedures, physical security of data and provision of detection of fraud and abuse.
- I as a software designer, worked really hard under following constraints: -
- **Hardware:** - The existing hardware will obviously affect the system design.
- **Software:** -The available software in the market will constraint the design.
- **Budget:** -The budget allocated for the project by the organization will affect the scope and depth of design.
- **Time Scale:** -The new system may be required by a particular time (may be by the start of the financial year). This may put a constraint on me to find the best design.

DESIGN METHODOLOGY

The framework for the new system developed during analysis guides the scope of the system design. More clearly defined logical method for developing system that meets user requirements has led to new techniques and methodologies that fundamentally attempt to do the following:

- Improve cementation and subsequent maintenance and enhancements.
- Cut down drastically on cost overruns and delays.
- Improve communication among the user.
- Simplify design by segmentation.

DESIGN PROCESS

The computer system design process is an exercise of specifying “how” the system will work. It is an iterative process which is base done “what” the system will do as shown in the feasibility report.

Mainly, following five parts have been included in the system design process: -

- **Output design:** The starting point of the design process is the proper knowledge of system requirements which will normally be converted in terms of output.
- **Input Design:** Once the output requirements have been finalized,

- the next step is to find out what data need to be made available to the system to produce the desire output.
- **File Design:** Once the input design is captured in the system, these may have to be preserved either for a short or long period. These data will generally be stored in files in a logical manner.
- **Procedures design:** This step involves specifications of how processing will be performed. In this, there are two aspects involves:
 - Computer procedure
 - Non –computer procedure

PROTOTYPE DESIGN

Prototype is a working system that is developed to test ideas of the concerned system and assumption about the new system. Like any computer-based system, it consists of working software that accepts input, performs some operation on it and gives the input.

The prototype is actually a pilot or test needed for my project. Information that I (as a system designer) gained through its use is applied to a modified design that may again be used as a prototype to reveal still better design information. I repeat the process as many times as necessary to reveal essential design requirement.

STRUCTURE DESIGN

This is the most important phase of designing as far as my project is concerned. It is a data flow-based methodology. I began this approach with a system specification that identifies input and output and describes the functional aspects of the system. Then, I used this very specification as a basis for the graphic representation. The next step that I took is the definition of the modules and their relationship to one another in a form called a structured chart, using a data dictionary and other structured tools.

INPUT DESIGN

The design of input play very significant role in getting the correct output. I follow certain guidelines for input design:

I kept in mind the idea while designing input that “effective design controls the quantity of data for input”. Data preparation and data entry operations depend on people. Since the cost of entering the data is high hence reducing data requirement means lowering cost, and it is the need of my organization to develop the software with minimum cost.

When processing is delayed owing to data preparation and data entry, the cause is called is called bottleneck. So, avoiding these anomalies was my aim.

Designing the source document and the input format:

The formats for the input records and the source document should be determined simultaneously, and I did perform the source task while working on the designing of the project.

The design of the documents should permit the personal recording the data to do so as easily and rapidly as possible. I use check boxes and some other utilities as and when required, which reduce the time needed to fill out documents and minimize recording errors. Actually, I designed the source document as soon as I come to know that what is actually needed and where and how it is to be entered into the system.

As I cite earlier that I designed input format concurrently with the source document. The factors that I considered for this are: record length, field size, use of codes and the relationship of the source document to the input record.

OUTPUT DESIGN

My Consideration before designing an output design

Type of user and purpose→ Generally different level of users will have different requirements from the system. So, I keep in mind the type of user that the said organization has to face and also the very purpose of my development.

Content→ The data that are to be included in the output. These will be related to the purpose of the output.

Format→ Here I considered the arrangement of data on the report, size of the paper, titles etc.

Frequency→ At what frequency (daily, weekly, monthly etc.) and when (after annual closing of accounts, after the end of the financial year, before the last day of every month etc.)

Quality→ This relates to the content, appearance and accuracy of the output.

FORM DESIGN & CONTROL

As we all know that data provide the basis for information system. Without data there is no system, but data must be fed in correct way so that the information produced must be in a format acceptable to the user. I also kept in mind while designing output screen that the screen should be user friendly because there is various type of people who are there to handle the software.

What is form?

People read from forms, write on forms and spend many hours in handling forms and filling forms. The data the form carry come from people, and the information output of the system goes to people. So, the form is a tool with a message. It is the physical carrier of data of information.

With this in my mind, it is hard for me to imagine a business operating without using forms. I designed the form of the said project user friendly

keeping in mind the people requirement.

Factors that I considered in form design

Forms design plays an important role in data processing. Forms must have the appearance of a well-conceived and attraction design. Some of the important factors that I have taken into consideration while designing output are as follows:

- Size and shape of the form should be such that it is convenient to handle.
- Arrange the material in a logical order so that it became easy to fill it up.
- The form title must clearly identify its purpose.
- Special feature like security and control are to be considered.

I as designer keep in mind that form should be designed in such a way so as to cover the specific needs of the purpose for which it is designed.

MODULES & THEIR DESCRIPTION

The task of the different modules depending upon the nature of work performed by it is as follows:

Login Module

This module is responsible to perform the login into the application.

Create user module

To login as customer, it is mandatory to create a customer account. In the login module the system will ask the data to be stored in the table.

Customer login

This module will ask username and password and will check it from the oracle Database. If the username and password are correct then it will login. Otherwise, it Will display the error message. After login, the customer can book a product and view his last transactions. He can also edit his profile.

Administrator login:

This module will ask username and password and will check it from the MYSQL Database and will check its privilege. If the username and password are correct and of the administrator then it will login. Otherwise, it will display the error message. Further it will redirect to the administrator page from where he can enter the product details, modify the price, stock etc.

Room:

Allow adding details of room, doctor and also allow deleting doctor record.

Admission:

Allow entering details of the patient.

Discharge:

Allow discharging in the patient according to room number and bed number.

Patient list:

Allow displaying the list of the patients such as patient name, room number and bed number.

Patient record:

Search the record of the patient on the basis of room number and bed number.

Edit:

Allow the facility to add rooms, doctor records and to delete doctor record

Quit:

Allow to exit from the entire software or system.

PROCESS LOGIC

The process logic of the project is very simple. The description of that is given below-

- The project begins with study & analysis of the activities of the school, staff members and attendant of the hospital.
- Then tables of the doctors, patient, treatment, students, employees, username-password, and payment etc. based on their relations are made.
- The whole process uses the concept of front-end & backend. Where JAVA and HTML is working as front-end and MYSQL 8.0 is working as backend.
- After that interfaces are designed in JAVA using forms and coding is done according to requirements of application.
- Finally we connect the front-end interface i.e.: JAVA application to the backend interface, which is the tables in MYQL. This connection will enable the access of database.

All required information would be displayed on the front-end application for the user.

- Use of the front-end & backend technology will enable to make the application reliable, fast & user friendly.
- In this way we go about the whole process of the design of the information system for the school.

The whole process logic can be shown with the help of a flow chart. The various steps can be seen clearly & accordingly the system can be made.

DATA INTEGRITY & CONSTRAINTS

Every system/Application must provide built-in features for security and integrity of data. Without safeguards against unauthorized access fraud, embezzlement etc a system could be so vulnerable as to threaten the survival of the organization.

System security refers to technical innovation and procedures applied to the hardware and the operating systems to protect against deliberate or accidental damage from defined threats. On the other hand data security is the protection of data from loss, disclosure, modification, and destruction. In this project, security is most important to prevent the program from the unauthorized user.

My Project "HOSPITAL MANAGEMENT SYSTEM" that starts from taking login and password from the user. Then verify it if found incorrect then again prompt from login and password. After the correct input it shows the welcome message and holds the message until any key is pressed. This project starts the work when the project correct entry is important at the time of i.e. id number, patient name, doctor name, doctor id, staff id number, staff name etc at the time of any types of queries, which is based on id number, name etc then wrong entries are invalid.

DATA BASE DESIGN

Database design is used to define and then specify the structure of business objects used in the client/server system. The analysis required to identify business objects is accomplished using business process engineering methods, conventional analysis modeling notation, such as ERD, can be used to define business objects, but a database repository should be established in order to capture the additional information that cannot be fully documented using a graphic notation such as an ERD.

In this repository, a business object is defined as information that is visible to the purchaser and user of the system. Not its implementers. This information, implemented using a relational database, can be maintained in a design repository. The following design information is collected for the client/server database.

- ✓ Entries are identified within the ERD for the new system.
- ✓ File implements the entries identified within the ERD.
- ✓ File to field relationships establish the layout for the files by identifying which fields are include in which files.
- ✓ Fields define the fields in the design (the data dictionary).
- ✓ File to file relationships identify related files that can be joined to create logical views or queries.

- ✓ Relationships validation identifies the type of file-to-file or file-to-field relationships used for validation.
- ✓ Field type is used to permit inheritance of field characteristics from filed super classes (e.g., date, text, number, value, price).
- ✓ Data type specifies the characteristics of the data contained in a field.
- ✓ Field functions include key, foreign key, attribute, virtual field, derived field and the like.
- ✓ Allowed values identify values allowed for the status type fields.
- ✓ Business rule are the rules for editing, calculation derived fields, and so on.

The trend towards distributed data management has accelerated as c/s architectures have become more pervasive. In c/s systems that implement this approach, the data management component resides on both the client and server. Within the context of database design, a key issue is data distribution. That is, how are data distributed between the client and server and dispersed across the nodes of a network?

A relational database system enables easy across to distribute data through the use of structured query language. The advantage of SQL in c/s architecture is that it is "no navigational" [BER92]. In RDBMS, the type of data is specified using SQL, but no navigational information is required. Of course, the implication of this is that the RDBMS must be sophisticated enough to maintain the location of all data and be capable of defining the

best path to it. In less sophisticated database system, a request for data must maintain navigational information, data management becomes much more complicated for c/s systems.

It should be noted that other data distribution and management technique are also available to the designer.

- **DATA STRUCTURES : =>** This is a cataloged storing the details of data being used in the system. Data dictionary serves the following purposes;-
 - Up gradation in the existing program
 - Maintenance which facilitates the effectiveness of the system
 - Data query

The various tables are as:

Login Table

Field Name	Type	Width	Constraints	Description
Uname	String	15	Primary Key	User Name
Upass	String	20	Not Null	Password

SignUp Table

Field Name	Type	Width	Constraints	Description
Uname	String	15	Primary Key	User Name
Upass	String	20	Not Null	Password
Name	String	20	Not Null	Person Name
sex	String	20	Not Null	Gender
Dob	Date	-	Not Null	Date of birth
Country	String	20	Not Null	Country
State	String	15	Not Null	State

HOSPITAL MANAGEMENT SYSTEM

City	String	20	Not Null	City
Street	String	20	Not Null	Street
Zip code	String	10	Not Null	Zip Code
MNo	String	12	Not Null	Mobile Number
e-mail	String	30	Not Null	Email Address
Status	String	15	Not Null	(active/suspended)
Type	String	15	Not Null	(user/administrator)

Hospital:

This table gives static information about hospital.

Field Name	Type	Width	Constraints	Description
HName	String	25	Not Null	Hospital Name
Location	String	30	Not Null	Hospital Address
Hid	String	15	Primary Key	Hospital Id Number

Doctor:

This table contains the information about hospital doctor.

Field Name	Type	Width	Constraints	Description
Did	String	25	Primary Key	Doctor Id Number
DDept	String	30	Not Null	Doctor Department
DName	String	15	Not Null	Doctor Name

Room:

This table contains the information about rooms allocated for patients.

Field Name	Type	Width	Constraints	Description
Roomno	String	25	Primary Key	Room Number
Reg_no	String	30	Not Null	Patient Registration Number
Bed_no	String	15	Not Null	Bed Number
Roomstatus	String	10	Not Null	Availability

Patient:

This table contains the information about patient.

Field Name	Type	Width	Constraints	Description
Reg_no	String	25	Primary Key	Patient Registration Number
Mrtstatus	String	20	Not Null	Marital Status
Sex	String	15	Not Null	Gender
P_address	String	15	Not Null	Patient Address
PName	String	20	Not Null	Patient Name

USER INTERFACE DESIGN

Interface design focuses on three areas of concern: (1) the design of interfaces between software components, (2) the design of interface between the software and other nonhuman producers and consumers of information (i.e. other external entities), and (3) the design of the interface between a human (i.e. the user) and the computer. In this project we focus exclusively on the third interfaces design category-*user interface design*.

User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and action and then creates a screen layout that forms the basis for a user interface prototype.

A software engineer designs the user interface by applying an iterative process that draws on widely accepted design principles. If software is difficult to use, if it forces us into mistake, or if it frustrates us efforts to

accomplish our goals, we won't like it, regardless of the computational power it exhibits or the functionality it offers. The interface has to be right because it molds a user's perception of the software.

User interface design begins with the identification of user, task and environmental requirements. Once user tasks have been identified, user scenarios are created and analyzed to define a set of interface objects and actions. These forms the basis for the creation of screen layout that depict graphical design and placement of icons, definition of descriptive screen text, specification and titling for windows, and specification of major and minor menu items. Tools are used to prototype and ultimately implement the design model, and the result is evaluated for quality.

User scenarios are created, and screen layouts are generated. An interface prototype is developed and modified in an iterative fashion.

The prototype is "test driven" by the users and feedback from the test drive is used for the next iterative modification of the prototype.

The process

The analysis & design process for user interface is iterative and can be represented using a spiral model in the figure given below-

The user interface analysis and design process encompasses four distinct framework activities.

- User, task, and environment analysis and modeling.

- Interface design.
- Interface construction (implementation).
- Interface validation.

The spiral shows in the given figure 1 implies that each of these task will occur more than once, with each pass around the spiral representing additional elaboration of requirements and the resultant design. In most cases, the construction activity involves prototyping – the only practical way to validate who will interact with system. Skill level, business understanding, and general receptiveness to the new system are recorded; and different user categories are defined. For each user category, requirements are elicited. In essence, the software engineers attempts to understand the system perception for each class of user.

Once general requirements have been defined, a more detailed task analysis is conducted. Those tasks that the user performs to accomplish the goals of the system are identified, described, and elaborated (over a number of iterative passes through the spiral).

The analysis of user requirement focuses on the physical work environment. Among the questions to be asked are:

- Where be the interface be located physically?
- Will the user be sitting, standing, or performing other tasks unrelated to the interface?

- Does the interface hardware accommodated space, light, or noise constraints?
- Are there special human factors consideration driven by environmental factors?

The information gathered as part of the analysis activity is used to create an analysis model for the interface. Using this model as a basic, the design activity commences.

The goals of interface design a set of interface objects and actions (and their screen representations) that enable a user to perform all defined task in a manner that meets every usability goal defined for the system the construction activity normally begins with the creation of the prototype that enables usage scenarios to be evaluated. As the interactive design process continues, user interface development tools may be used to complete the interface of the interface.

Validation focuses on (1) the ability of the interface to implement every task correctly, to accommodate all task variations, and to achieve all general user requirements; (2) the degree t

As we have already noted, the activities described in this section occur iteratively. Therefore, there is no need to attempt to specify every detail (for the analysis or design model) on the first pass. Subsequently passes through the process elaborated task detail, design information, and the operational features of the interface.

CODING

AddDoctor.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>ADD DOCTOR</title>
</head>
<body>
<form name="f1" method="get" action=" ../JspFile/readadddoctor.jsp">
<h1><Font size="+3" color="red">ENTER DOCTOR DETIALS</font></h1>
<table>
<tr>
<td>Doctor Name</td>
<td><input type="text" name="Dn" value="" size="20"
placeholder="doctorname"></td>
</tr>
<tr>
<td>Father name</td>
<td><input type="text" name="fn" value="" size="20" placeholder="fathername"></td>
</tr>
<tr>
<td>GENDER</td>
<td><input type="radio" id="male" name="gn" value="male" checked>Male
<input type="radio" id="femlae" name="gn" value="female">Female
<input type="radio" id="other" name="gn" value="other">Other</td>
</tr>
<tr>
<td>DATE OF BIRTH</td>
<td>
<select name="db" size="1">
<option value="MON" selected>DD</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
```

```
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select name="mn" size="1">
<option value="MON" selected>MON</option>
<option value="01">JAN</option>
<option value="02">FEB</option>
<option value="03">MAR</option>
<option value="04">APR</option>
<option value="05">MAY</option>
<option value="06">JUN</option>
<option value="07">JUL</option>
<option value="08">AUG</option>
<option value="09">SEP</option>
<option value="10">OCT</option>
<option value="11">NOV</option>
<option value="12">DEC</option>
</select>
<select name="yr" size="1">
<option value="YYYY" selected>YYYY</option>
<option value="1975">1975</option>
<option value="1976">1976</option>
<option value="1977">1977</option>
<option value="1978">1978</option>
<option value="1979">1979</option>
<option value="1980">1980</option>
<option value="1981">1981</option>
<option value="1982">1982</option>
<option value="1983">1983</option>
<option value="1984">1984</option>
<option value="1985">1985</option>
<option value="1986">1986</option>
```

```

<option value="1987">1987</option>
<option value="1988">1988</option>
<option value="1989">1989</option>
<option value="1990">1990</option>
<option value="1991">1991</option>
<option value="1992">1992</option>
<option value="1993">1993</option>
<option value="1994">1994</option>
<option value="1995">1995</option>
<option value="1996">1996</option>
<option value="1997">1997</option>
<option value="1998">1998</option>
<option value="1999">1999</option>
<option value="2000">2000</option>
<option value="2001">2001</option>
<option value="2002">2002</option>
<option value="2003">2003</option>
<option value="2004">2004</option>
<option value="2005">2005</option>
<option value="2006">2006</option>
<option value="2007">2007</option>
<option value="2008">2008</option>
<option value="2009">2009</option>
<option value="2010">2010</option>
<option value="2011">2011</option>
<option value="2012">2012</option>
<option value="2013">2013</option>
<option value="2014">2014</option>
<option value="2015">2015</option>
<option value="2016">2016</option>
<option value="2017">2017</option>
<option value="2018">2018</option>
<option value="2019">2019</option>
<option value="2020">2020</option>
<option value="2021">2021</option>
<option value="2022">2022</option>
</select>
</tr>
<tr>
<td>ADDRESS</td>
<td><input type="text" name="ad" value="" size="20" placeholder="address"></td>
</tr>
<tr>
<td>QUALIFICATION</td>
<td><input type="text" name="qf" value="" size="20"
placeholder="qualification"></td>

```

```

</tr>
<tr>
<td>EXPERIANCE</td>
<td><input type="text" name="ex" value="" size="20" placeholder="experiance"></td>
</tr>
<tr>
<td>LAST WORKED HOSPITAL</td>
<td><input type="text" name="lwh" value="" size="20"
placeholder="lastworkedhospital"></td>
</tr>
<tr>
<td>SALARY</td>
<td><input type="text" name="sa" value="" size="20" placeholder="salary"></td>
</tr>
<tr>
<td></td>
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</table>
</form>
</body>
</html>

```

AddPatient.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>ADD PATIENT</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readaddpatient.jsp">
<h1><Font size="+3" color="red">ENTER PATIENT DETAILS</font></h1>
<table>
<tr>
<td>Patient Name</td>
<td><input type="text" name="pn" value="" size="20"
placeholder="patientname"></td>
</tr>
<tr>
<td>Age</td>
<td><input type="text" name="ag" value="" size="20" placeholder="age"></td>
</tr>
<tr>
<td>Gender</td>
<td><input type="radio" id="male" name="gn" value="male">Male
<input type="radio" id="femlae" name="gn" value="female">Female

```



```

<input type="radio" id="other" name="gn" value="other">Other</td>
</tr>
<tr>
<td>Address</td>
<td><input type="text" name="ad" value="" size="20" placeholder="Address1"></td>
</tr>
<tr>
<td>Consult doctor</td>
<td><input type="text" name="cd" value="" size="20"
placeholder="consultdoctor"></td>
</tr>
<tr>
<td>Problem</td>
<td><input type="text" name="pr" value="" size="20" placeholder="problem"></td>
</tr>
<tr>
<td>Admitted</td>
<td><input type="text" name="add" value="" size="20" placeholder="Admitted"></td>
</tr>
<tr>
<td>Non admitted</td>
<td><input type="text" name="na" value="" size="20"
placeholder="nonadmitted"></td>
</tr>
<tr>
<td>Fee</td>
<td><input type="text" name="fe" value="" size="20" placeholder="fee"></td>
</tr>
<tr>
<td>Diagnosis</td>
<td><input type="text" name="di" value="" size="20" placeholder="diagnosis"></td>
</tr>
<tr>
<td>Mobile No.</td>
<td><input type="text" name="mo" value="" size="20" placeholder="mobile.no"></td>
</tr>
<tr>
<td>Contact number</td>
<td><input type="text" name="cn" value="" size="20"
placeholder="contactnumber"></td>
</tr>
<tr>
<td>Email.Id</td>
<td><input type="email" name="em" value="" size="20" placeholder="email.id"></td>
</tr>
<tr>

```

```
<td></td>
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</table>
</form>
</body>
</html>
```

AddWard.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>ADD WARD</title>
</head>
<form name="f1" method="get" action="../JspFile/readaddward.jsp">
<h1><font size="+3" color=red>ADD WARD DETAILS </font></h1>
<table>
<tr>
<th height="34">WARD NAME</th>
<td><input type="text" name="wn" value="" size="15"placeholder="Wardname"></td>
</tr>
<tr>
<th>WARD TYPE</th>
<td><select name="qu" id="TYPE">
<option value=" ">Choose option</option>
<option value="Pediatrics">Pediatrics</option>
<option value="Maternity ">Maternity</option>
<option value="Psychiatric ">Psychiatric</option>
<option value="Geriatrics ">Geriatrics</option>
</select>
</td>
</tr>
<tr>
<th height="33">WARD MODE</th>
<td><input type="text" name="wm" value="" size="15" placeholder="Ward
mode"></td>
</tr>
<tr>
<th height="33">TOTAL BED</th>
<td><input type="text" name="tb" value="" size="15" placeholder="Total Bed"></td>
</tr>
<tr>
<th height="33">COST</th>
<td><input type="text" name="rp" value="" size="15" placeholder="Amount"></td>
</tr>
<tr>
```

```
<th height="33">ROOM TYPE</th>
<td><input type="radio" id="ac" name="rt" value="ac">AC Room<br>
<input type="radio" id="Non-ac" name="rt" value="Non-ac">Non-ac Room
</td>
</tr>
</table>
<br>
<input type="submit" value="Submit">
<input type="submit" value="Reset">
</form>
</body>
</html>
```

Attendance.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>ATTENDANCE</title>
</head>
<body bgcolor="silver">
<center>
<form name="f1" method="get" action="../JspFile/readattendance.jsp">
<h1><font size="+3" color="red">ENTER ATTENDANCE</font></h1>
<table>
<tr>
<td>Date Of Attendance</td>
<td>
<select name="db" size="1">
<option value="DD" selected>DD</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
```

```
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select name="mn" size="1">
<option value="MON" selected>MON</option>
<option value="01">JAN</option>
<option value="02">FEB</option>
<option value="03">MAR</option>
<option value="04">APR</option>
<option value="05">MAY</option>
<option value="06">JUN</option>
<option value="07">JUL</option>
<option value="08">AUG</option>
<option value="09">SEP</option>
<option value="10">OCT</option>
<option value="11">NOV</option>
<option value="12">DEC</option>
</select>
<select name="yr" size="1">
<option value="YYYY" selected>YYYY</option>
<option value="1975">1975</option>
<option value="1976">1976</option>
<option value="1977">1977</option>
<option value="1978">1978</option>
<option value="1979">1979</option>
<option value="1980">1980</option>
<option value="1981">1981</option>
<option value="1982">1982</option>
<option value="1983">1983</option>
<option value="1984">1984</option>
<option value="1985">1985</option>
<option value="1986">1986</option>
<option value="1987">1987</option>
<option value="1988">1988</option>
```

```

<option value="1989">1989</option>
<option value="1990">1990</option>
<option value="1991">1991</option>
<option value="1992">1992</option>
<option value="1993">1993</option>
<option value="1994">1994</option>
<option value="1995">1995</option>
<option value="1996">1996</option>
<option value="1997">1997</option>
<option value="1998">1998</option>
<option value="1999">1999</option>
<option value="2000">2000</option>
<option value="2001">2001</option>
<option value="2002">2002</option>
<option value="2003">2003</option>
<option value="2004">2004</option>
<option value="2005">2005</option>
<option value="2006">2006</option>
<option value="2007">2007</option>
<option value="2008">2008</option>
<option value="2009">2009</option>
<option value="2010">2010</option>
<option value="2011">2011</option>
<option value="2012">2012</option>
<option value="2013">2013</option>
<option value="2014">2014</option>
<option value="2015">2015</option>
<option value="2016">2016</option>
<option value="2017">2017</option>
<option value="2018">2018</option>
<option value="2019">2019</option>
<option value="2020">2020</option>
<option value="2021">2021</option>
<option value="2022">2022</option>
</select>
</tr>
<tr>
<td>Incoming time</td>
<td><input type="text" name="it" value="" size="20"
placeholder="incomingtime"></td>
</tr>
<tr>
<td>Outgoing time</td>
<td><input type="text" name="ot" value="" size="20"
placeholder="outgoingtime"></td>
</tr>

```

```

<tr>
<td>Thank involved</td>
<td><input type="text" name="ti" value="" size="20"
placeholder="thankinvolved"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit">
<input type="reset" value="reset"></td>
</table>
</form>
</center>
</body>
</html>

```

Bill.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>BILL</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readbill.jsp">
<h1><Font size="+3" color="red">ENTER BILL </Font></h1>
<table>
<tr>
<td>Patient Id number</td>
<td><input type="text" name="pid" value="" size="20"
placeholder="patientidnumber"></td>
</tr>
<tr>
<td>Patient name</td>
<td><input type="text" name="pn" value="" size="20"
placeholder="patientname"></td>
</tr>
<tr>
<td>Consult doctor</td>
<td><input type="text" name="cd" value="" size="20"
placeholder="consultdoctor"></td>
</tr>
<tr>
<td>Diagnosis</td>
<td><input type="text" name="di" value="" size="20" placeholder="dianosis"></td>
</tr>
<tr>
<td>CONTACT NUMBER</td>

```

```

<td><input type="text" name="cn" value="" size="20"
placeholder="contactnumber"></td>
</tr>
<tr>
<td>Mobile number</td>
<td><input type="text" name="mo" value="" size="20"
placeholder="mobilenumber"></td>
</tr>
<tr>
<td>Email.Id</td>
<td><input type="text" name="em" value="" size="20" placeholder="email.id"></td>
</tr>
<tr>
<td>Address</td>
<td><input type="text" name="ad" value="" size="20" placeholder="address"></td>
</tr>
<tr>
<td></td>
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</table>
</form>
</body>
</html>

```

ChangePassword.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Change Password</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readchangepassword.jsp">
<h1><font size="+3" color=red>WELCOME TO HOSPITAL </font></h1>
<table >
<tr>
<th height="34">Enter your user name </th>
<td><input type="text" name="un" value="" size="20"placeholder="Username"></td>
</tr>
<tr>
<th height="33">Enter Current Password</th>
<td><input type="password" name="cp" value="" size="20" placeholder="Current
Password"></td>
</tr>
<tr>
<th height="33">Enter New Password</th>

```

```

<td><input type="password" name="np" value="" size="20" placeholder="New
Password"></td>
</tr>
<tr>
<th height="33">Retype New Password</th>
<td><input type="password" name="rp" value="" size="20" placeholder="Retype
Password"></td>
</tr>
<tr>
<th height="39" colspan="2">
<input type="submit" value="Login">
<input type="submit" value="cancel"></th>
</tr>
</table>
</form>
</body>
</html>

```

Contactus.html

```

<html>
<head>
<title>contact us</title>
</head>
<body bgcolor="silver">
<form name="f1" method="get" action="../JspFile/readcontact.jsp">
<center>
<h1><font size="+3" color="red">CONTACT US</font></h1>
<table>
<tr>
<td><input type="email" name="Dn" value="" size="30"
placeholder="email"></td><br>
</tr>
<tr>
<td><input type="text" name="Dn" value="" size="30"
placeholder="subject"><br></td>
</tr>
</table>
</center>
</form>
</body>
</html>

```

DeleteAttendance.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>DELETE ATTENDANCE</title>

```



```
</head>
<body>
<form name="f1" method="get" action="../JspFile/readdeleteattendance.jsp">
<h1><Font size="+3" color="red">Delete attendance </Font></h1>
<table>
<tr>
<td>Id number</td>
<td><input type="text" name="id" value="" size="20" placeholder="idnumber"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit">
<input type="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>
```

DeleteDoctor.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>DELETE DOCTOR</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readdeletedoctor.jsp">
<h1><Font size="+3" color="red">ENTER DELETE DOCTOR</font></h1>
<table>
<tr>
<td>DOCTOR ID</td>
<td><input type="text" name="di" value="" size="20" placeholder="doctorid"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit">
<input type="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>
```

DeletePatient.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>DELETE PATIENT</title>
```

```
</head>
<body>
<form name="f1" method="get" action="../JspFile/readdeletepatient.jsp">
<h1><Font size="+3" color="red">Delete patient </Font></h1>
<table>
<tr>
<td>Patient number</td>
<td><input type="text" name="pn" value="" size="20"
placeholder="patientnumber"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit">
<input type="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>
```

DeleteWard.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Delete ward</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readdeleteward.jsp">
<h1><Font size="+3" color="red">ENTER DELETE WARD</font></h1>
<table>
<tr>
<td>Ward number</td>
<td><input type="text" name="wn" value="" size="20"
placeholder="wardnumber"></td>
</tr>
<tr>
<td></td>
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</tr>
</table>
</form>
</body>
</html>
```

Feedback.html

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="ISO-8859-1">
<title>Feedback</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readfeedback.jsp">
<h1><Font size="+3" color=red>Enter Feedback Details </font></h1>
<table>
<tr>
<th height="34">user name</th>
<td><input type="text" name="un" value="" size="20"placeholder="username"></td>
</tr>
<tr>
<th height="33">EMAIL ID</th>
<td><input type="email" name="ei" value="" size="20" placeholder="email"></td>
</tr>
<tr>
<th height="39" colspan="2">
<input type="submit" value="Login">
<input type="submit" value="cancel"></th>
</tr>
</table>
</form>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login Page</title>
</head>
<body>
<form name="f1" method="get" action="../JspFile/readlogin.jsp">
<h1><Font size="+3" color=red>ENTER LOGIN DETAILS </font></h1>
<table >
<tr>
<th height="34">Enter your username</th>
<td><input type="text" name="un" value="" size="20"placeholder="username"></td>
</tr>
<tr>
<th height="33">Enter your password</th>
<td><input type="password" name="pw" value="" size="20"
placeholder="password"></td>
</tr>
<tr>
<th height="39" colspan="2">
```

```
<input type="submit" value="Login">
<input type="submit" value="cancel"></th>
</tr>
</table>
</form>
</body>
</html>
```

Forgotpassword.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>ForgetPassowrd</title>
</head>
<body>
<form name="f1" method="get" action=" ../JspFile/readforgetpassword.jsp">
<h1><Font size="+3" color=red>ENTER FORGETPASSWORD DETAILS
</font></h1>
<table>
<tr>
<td>Enter your user name</td>
<td><input type="text" name="un" value="" size="20"placeholder="Username"></td>
</tr>
<tr>
<td>Email.Id</td>
<td><input type="email" name="em" value="" size="20" placeholder="Email.Id"></td>
</tr>
<tr>
<td>Secret Question</td>
<td><select name="qu">
<option value=" ">Choose option
<option value=" ">Enter your Pet Name
<option value=" ">Enter your Birth Place
<option value=" ">Enter your friend Name
</select>
</td>
</tr>
<tr>
<td>Answer</td>
<td><input type="text" name="an" value="" size="20" placeholder="Your
Answer"></td>
</tr>
<tr>
<th height="39" colspan="2">
<input type="submit" value="Submit">
<input type="submit" value="Reset"></th>
```

```
</tr>
</table>
</form>
</body>
</html>
```

ModifyAttendance.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>MODIFY ATTENDANCE</title>
</head>
<body>
<form name="f1" method="get" action="../../JspFile/readmodifyattendance.jsp">
<h1><Font size="+3" color="red">Modify attendance </Font></h1>
<table>
<tr>
<td>Id number</td>
<td><input type="text" name="id" value="" size="20" placeholder="idnumber"></td>
</tr>
<tr>
<td></td>
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</tr>
</table>
</form>
</body>
</html>
```

ModifyPatient.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>MODIFY PATIENT</title>
</head>
<body>
<form name="f1" method="get" action="../../JspFile/readmodifypatient.jsp">
<h1> <Font size="+3" color="red">ENTER PATIENT DETIALS</Font></h1>
<table>
<tr>
<td>Patient number</td>
<td><input type="text" name="pn" value="" size="20"
placeholder="patientnumber"></td>
</tr>
<tr>
<td></td>
```

```
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</table>
</form>
</body>
</html>
```

ModifyWard.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>MODIFY WARD</title>
<body>
</head>
<form name="f1" method="get" action="../JspFile/readmodifyward.jsp">
<h1><font size="+3" color="red">ENTER MODIFY WARD</font></h1>
<table>
<tr>
<td>Ward number</td>
<td><input type="text" name="wn" value="" size="20"
placeholder="wardnumber"></td>
</tr>
<tr>
<td></td>
<td><input type=submit value=submit>
<input type=reset value=reset></td>
</tr>
</table>
</form>
</body>
</html>
```

SignUp.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title></title>
</head>
<body bgcolor="silver">
<form name="f1" method="get" action="../JspFile/readsignup.jsp">
<center>
<h1><font size="+3" color="red">ENTER YOUR DETIALS </font></h1>
<table>
<tr>
<td>User Name</td>
<td><input type="text" name="un" value="" size="20" placeholder="username"></td>
</tr>
```

```
<tr>
<td>Password</td>
<td><input type="password" name="pw" value=""
size="20"placeholder="Password"></td>
</tr>
<tr>
<td>RetypePassword</td>
<td><input type="password" name="pw1" value=""
size="20"placeholder="Password"></td>
</tr>
<tr>
<td>First Name</td>
<td><input type="text" name="fn" value="" size="20" placeholder="first name"></td>
</tr>
<tr>
<td>Middle Name</td>
<td><input type="text" name="ma" value="" size="20" placeholder="middle
name"></td>
</tr>
<tr>
<td>Last Name</td>
<td><input type="text" name="ln" value="" size="20" placeholder="last name"></td>
</tr>
<tr>
<td>DATE OF BIRTH</td>
<td>
<select name="db" size="1">
<option value="DD" selected>DD</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
```

```
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select name="mn" size="1">
<option value="MON" selected>MON</option>
<option value="01">JAN</option>
<option value="02">FEB</option>
<option value="03">MAR</option>
<option value="04">APR</option>
<option value="05">MAY</option>
<option value="06">JUN</option>
<option value="07">JUL</option>
<option value="08">AUG</option>
<option value="09">SEP</option>
<option value="10">OCT</option>
<option value="11">NOV</option>
<option value="12">DEC</option>
</select>
<select name="yr" size="1">
<option value="YYYY" selected>YYYY</option>
<option value="1975">1975</option>
<option value="1976">1976</option>
<option value="1977">1977</option>
<option value="1978">1978</option>
<option value="1979">1979</option>
<option value="1980">1980</option>
<option value="1981">1981</option>
<option value="1982">1982</option>
<option value="1983">1983</option>
<option value="1984">1984</option>
<option value="1985">1985</option>
<option value="1986">1986</option>
<option value="1987">1987</option>
<option value="1988">1988</option>
<option value="1989">1989</option>
```



```

<option value="1990">1990</option>
<option value="1991">1991</option>
<option value="1992">1992</option>
<option value="1993">1993</option>
<option value="1994">1994</option>
<option value="1995">1995</option>
<option value="1996">1996</option>
<option value="1997">1997</option>
<option value="1998">1998</option>
<option value="1999">1999</option>
<option value="2000">2000</option>
<option value="2001">2001</option>
<option value="2002">2002</option>
<option value="2003">2003</option>
<option value="2004">2004</option>
<option value="2005">2005</option>
<option value="2006">2006</option>
<option value="2007">2007</option>
<option value="2008">2008</option>
<option value="2009">2009</option>
<option value="2010">2010</option>
<option value="2011">2011</option>
<option value="2012">2012</option>
<option value="2013">2013</option>
<option value="2014">2014</option>
<option value="2015">2015</option>
<option value="2016">2016</option>
<option value="2017">2017</option>
<option value="2018">2018</option>
<option value="2019">2019</option>
<option value="2020">2020</option>
<option value="2021">2021</option>
<option value="2022">2022</option>
</select>
</tr>
<tr>
<td>Gender</td>
<td><input type="radio" id="male" name="gn" value="male">Male
<input type="radio" id="femlae" name="gn" value="female">Female
<input type="radio" id="other" name="gn" value="other">Other</td>
</tr>
<tr>
<td>Aadhar No.(Optional)</td>
<td><input type="text" name="aa" value="" size="20" placeholder="aadhar"></td>
</tr>
<tr>

```

```

<td>Mobile No.</td>
<td><input type="text" name="mo" value="" size="20" placeholder="mobile.no"></td>
</tr>
<tr>
<td>Email.Id</td>
<td><input type="email" name="em" value="" size="20" placeholder="email.id"></td>
</tr>
<tr>
<td>Secret Question</td>
<td><select name="qu" id="question">
<option value="">Choose option</option>
<option value="Enter your Pet Name">Enter your Pet Name</option>
<option value="Enter your Birth Place">Enter your Birth Place</option>
<option value="Enter your friend Name">Enter your friend Name</option>
</select>
</td>
</tr>
<tr>
<td>Answer</td>
<td><input type="text" name="an" value="" size="20" placeholder="Your
Answer"></td>
</tr>
<tr>
<td>Address</td>
<td><input type="text" name="ad" value="" size="20" placeholder="Address1"></td>
</tr>
<tr>
<td>Country Name</td>
<td><input type="text" name="cn"></td>
</tr>
<tr>
<td>zip Code</td>
<td><input type="text" name="zc"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit">
<input type="reset" value="reset"></td>
</tr>
</table>
</center>
</form>
</body>
</html>
UpdateDoctor.html
<!DOCTYPE html>
<html>

```

```
<head>
<meta charset="ISO-8859-1">
<title>UPDATE DOCTOR</title>
</head>
<body>
<form name="f1" method="get" action=" ../JspFile/readupdatedoctor.jsp">
<h1><font size="+3" color="red"> ENTER UPDATE DOCTOR </font></h1>
<table>
<tr>
<td>DOCTOR ID</td>
<td><input type="text" name="di" value="" size="20" placeholder="doctorid"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit">
<input type="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>
```

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" %>
<html>
<head>
<meta http-equiv="content-Type" content="text/html; charset=ISO-8859-1">
<title>Hospital Management system</title>
</head>
<body>
<table>
<tr>
<td><a href=" ../HtmlFile/Adddoctor.html">Adddoctor</a></td>
</tr>
<tr>
<td><a href=" ../HtmlFile/Deletedoctor.html">Delete doctor</a></td>
</tr>
</table>
</body>
```

readadddoctor.jsp

```
<%@ page import="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
```

```
String username="root";
String password="12345";
String dn=request.getParameter("Dn");
String fn=request.getParameter("fn");
String gn=request.getParameter("gn");
String db=request.getParameter("db");
String mn=request.getParameter("mn");
String yr=request.getParameter("yr");
String dob=yr+"-"+mn+"-"+db;
String ad=request.getParameter("ad");
String qf=request.getParameter("qf");
String ex=request.getParameter("ex");
int expr=Integer.parseInt(ex);
String Lwh=request.getParameter("lwh");
String sa=request.getParameter("sa");
int sal=Integer.parseInt(sa);
Random r=new Random();
int did=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into doctor values("+did+", '"+dn+"', '"+fn+"', '"+gn+"', '"+
dob+"', '"+ad+"', '"+qf+"', '"+expr+"', '"+Lwh+"', '"+sal+"')";
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1><font color=red> Record Inserted Successfully</font></h1>");
}
else
{
out.print("<h1><font color=red> Sorry! Record Could Not be Inserted</font></h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
```

```
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
readaddpatient.jsp
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String pn=request.getParameter("pn");
String ag=request.getParameter("ag");
int age=Integer.parseInt(ag);
String gn=request.getParameter("gn");
String ad=request.getParameter("ad");
String cd=request.getParameter("cd");
String pr=request.getParameter("pr");
String add=request.getParameter("add");
String na=request.getParameter("na");
String fe=request.getParameter("fe");
int fee=Integer.parseInt(fe);
String di=request.getParameter("di");
String mo=request.getParameter("mo");
String cn=request.getParameter("cn");
String em=request.getParameter("em");
Random r=new Random();
int pid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
```

```
String sql="insert into addpatient values("+pid+", "+pn+", "+age+", "+gn+", "+ad
+ ", "+cd+", "+pr+", "+add+", "+na+", "+fee+", "+di+", "+mo+", "+cn+", "+em+)";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
readaddward.jsp
<%@ page import = "java.sql.* , java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String wn=request.getParameter("wn");
String qu=request.getParameter("qu");
String wm=request.getParameter("wm");
String tb=request.getParameter("tb");
```

```
int tbed=Integer.parseInt(tb);
String rp=request.getParameter("rp");
int cost=Integer.parseInt(rp);
String rt=request.getParameter("rt");
Random r=new Random();
int wid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into addward values("+wid+", "+wn+", "+
qu+", "+wm+", "+tb+", "+rp+", "+rt+"";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
```

readaddattendance.jsp

```
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";

String db=request.getParameter("db");
String mn=request.getParameter("mn");
String yr=request.getParameter("yr");
String doa=yr+"-"+mn+"-"+db;
String it=request.getParameter("it");
int incomtym =Integer.parseInt(it);
String ot=request.getParameter("ot");
int outgoingtym =Integer.parseInt(ot);
String ti=request.getParameter("ti");
Random r=new Random();
int idno=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into attendance values('"+idno+"','"+doa+"','"+it+"','"+ot+"','"+ti+"')";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
}
```



```
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
```

readbill.jsp

```
<%@ page import = "java.sql.* , java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
```

```
<%
```

```
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String pin=request.getParameter("pin");
String pn=request.getParameter("pn");
String cd=request.getParameter("cd");
String di=request.getParameter("di");
String cn=request.getParameter("cn");
String mo=request.getParameter("mo");
String em=request.getParameter("em");
String ad=request.getParameter("ad");
Random r=new Random();
int bid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into bill values("+bid+", "+pin+", "+pn+", "+cd
+", "+di+", "+cn+", "+mo+", "+em+", "+ad+")";
out.print("<p>Sql="+sql);
```

```
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
readchangepassword.jsp
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String un=request.getParameter("un");
String cp=request.getParameter("cp");
String np=request.getParameter("np");
String rp=request.getParameter("rp");
Random r=new Random();
int cid=Math.abs(r.nextInt())/100000;
out.print("<center>");
```

```
out.println("<h1><u>Password Details</u></h1></center>");
out.print("<br>Your Uername= "+un);
out.print("<br>Your Old Passowrd= "+cp);
out.print("<br>Your New Passowrd= "+np);
out.print("<br>Conform Passowrd= "+rp);
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into changepassword values("+cid+", "+
un+", "+cp+", "+np+", "+rp+")";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
```

readdeleteattendance.jsp

```
<%@ page import ="java.sql.*" language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
ResultSet rs=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String id=request.getParameter("id");
int idno=Integer.parseInt(id);
out.print("<center>");
out.print("<p>Id Number="+id+"</p>");
out.print("<br>Id number = "+id);
int idno1=0;
int x=0;
String thnk="", doa="";
int didnumber=0,incomingtym=0, outgoingtym=0;
String sql="";
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
sql="select * from attendance";
rs=st.executeQuery(sql);
while(rs.next())
{
idno1=rs.getInt("idno");
if(idno==idno1)
{
doa=rs.getString("doa");
incomingtym=rs.getInt("incomingtym");
outgoingtym=rs.getInt("outgoingtym");
thnk=rs.getString("thnkinvol");
x=1;
break;
}
}
if(x==1)
{
```

```
sql="insert into dattendance values("+idno+", "+"+doa+", "+"+incomingtym
+", "+"+outgoingtym+", "+"+thnk+"");
out.print("<p>" + sql);
st.executeUpdate(sql);
sql="delete from attendance where idno="+idno;
out.print("<p>" + sql);
st.executeUpdate(sql);
out.print("<h1><center><font color=blue>Record Deleted
Successfully</font></center><h1>");
}
else
{
out.print("<h1><center><font color=red>Sorry! Record Not
Found</font></center><h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
}%>
readdeletedoctor.jsp
<%@ page import ="java.sql.*" language="java" contentType="text/html; charset=ISO-
8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
ResultSet rs=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
```

```
String di=request.getParameter("di");
int did=Integer.parseInt(di);
int didnumber=0;
int x=0;
String dname="", dob="", fname="",gender="",address="",qualif="",lastwhos="";
int exper =0,salary=0;
String sql="";
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
sql="select * from doctor";
rs=st.executeQuery(sql);
while(rs.next())
{
did=rs.getInt("did");
if(did==did)
{
dname=rs.getString("dname");
fname=rs.getString("fname");
gender=rs.getString("gender");
dob=rs.getString("dob");
address=rs.getString("address");
qualif=rs.getString("qualif");
exper=rs.getInt("exper");
lastwhos=rs.getString("lastwhos");
salary=rs.getInt("salary");
x=1;
break;
}
}
if(x==1)
{
sql="insert into doctor values("+did+", "+fname+", "+dname+", "+gender+
+", "+dob+", "+address+", "+qualif+", "+exper+", "+lastwhos+", "+salary+)";
out.print("<p>"+sql);
st.executeUpdate(sql);
sql="delete from doctor where did="+did;
out.print("<p>"+sql);
st.executeUpdate(sql);
out.print("<h1><center><font color=blue>Record Deleted
Successfully</font></center><h1>");
```

```
}
else
{
out.print("<h1><center><font color=red>Sorry! Record Not
Found</font></center><h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
}%>
```

readdeletepatient.jsp

```
<%@ page import ="java.sql.*" language="java" contentType="text/html; charset=ISO-
8859-1" pageEncoding="ISO-8859-1"%>
```

```
<%
```

```
Connection con=null;
```

```
Statement st=null;
```

```
ResultSet rs=null;
```

```
String driver="com.mysql.jdbc.Driver";
```

```
String url="jdbc:mysql://localhost:3306/hospital";
```

```
String username="root";
```

```
String password="12345";
```

```
String pn=request.getParameter("pn");
```

```
int pidnumber=Integer.parseInt(pn);
```

```
String pname="", gender="",
```

```
address="",condoc="",prob="",admi="",nonadmi="",diag="",mobilenumber="",contnum
ber="",emailid="";
```

```
int pid=0,age=0,fee=0;
```

```
int x=0;
```

```
try
```

```
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="select * from addpatient";
rs=st.executeQuery(sql);
while(rs.next())
{
pid=rs.getInt("pid");
if(pid==pid)
{
pname=rs.getString("pname");
age=rs.getInt("age");
gender=rs.getString("gender");
condoc=rs.getString("condoc");
prob=rs.getString("prob");
admi=rs.getString("admi");
nonadmi=rs.getString("nonadmi");
fee=rs.getInt("fee");
diag=rs.getString("diag");
mobilenumber=rs.getString("mobilenumb");
contnumber=rs.getString("contnumb");
emailid=rs.getString("emailid");
x=1;
break;
}
}
if(x==1)
{
sql="insert into addpatient values("+pid+", '"+pname+"', "+age+", '"+gender+"', '"+
address+"', '"+condoc+"', '"+prob+"', '"+admi+"', '"+nonadmi+"', "+fee+", '"+diag+"', '"+mobil
enumber+"', '"+contnumber+"', '"+emailid+"')";
out.print("<p>"+sql);
st.executeUpdate(sql);
sql="delete from addpatient where pid="+pid;
out.print("<p>"+sql);
st.executeUpdate(sql);
out.print("<h1><center><font color=blue>Record Deleted
Successfully</font></center><h1>");
}
else
{

```



```
out.print("<h1><center><font color=red>Sorry! Record Not  
Found</font></center><h1>");  
}  
}  
catch(Exception e)  
{  
out.println("<p>Error Arises="+e.getMessage());  
}  
finally  
{  
try  
{  
if(st!= null)  
st.close();  
if(con!=null)  
con.close();  
}  
catch(Exception e)  
{  
out.println("<p>Error Arises");  
}  
}  
%>
```

readdeleteward.jsp

```
<%@ page import = "java.sql.*" language="java" contentType="text/html; charset=ISO-  
8859-1" pageEncoding="ISO-8859-1"%>
```

```
<%
```

```
Connection con=null;  
Statement st=null;  
ResultSet rs=null;  
String driver="com.mysql.jdbc.Driver";  
String url="jdbc:mysql://localhost:3306/hospital";  
String username="root";  
String password="12345";  
String wn=request.getParameter("wn");  
int widnumber=Integer.parseInt(wn);  
String wname="", wtype="", wmode="", rtype="";  
int wid=0, tbed=0, cost=0;  
int x=0;  
try  
{  
Class.forName(driver);  
out.println("<p>Driver Registered");  
con=DriverManager.getConnection(url,username,password);  
out.println("<p>Connection Established");  
st=con.createStatement();
```

```
out.println("<p>Statement Created");
String sql="select * from addward";
rs=st.executeQuery(sql);
while(rs.next())
{
wid=rs.getInt("wid");
if(wid==wid)
wname=rs.getString("wname");
wtype=rs.getString("wtype");
wmode=rs.getString("wmode");
tbed=rs.getInt("tbed");
cost=rs.getInt("cost");
rtype=rs.getString("rtype");
x=1;
break;
}
if(x==1)
{
sql="insert into addward values("+wid+", "+wname+", "+wtype+", "+
wmode+", "+tbed+", "+cost+", "+rtype+")";
out.print("<p>" +sql);
st.executeUpdate(sql);
sql="delete from addward where wid="+wid;
out.print("<p>" +sql);
st.executeUpdate(sql);
out.print("<h1><center><font color=blue>Record Deleted
Successfully</font></center><h1>");
}
else
{
out.print("<h1><center><font color=red>Sorry! Record Not
Found</font></center><h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
```

```
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
readfeedback.jsp
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String un=request.getParameter("un");
String ei=request.getParameter("ei");
Random r=new Random();
int fid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into feedback values("+fid+", '"+un+"', '"+ei+"'");
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
```

```
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
}%>
```

readforgotpassword.jsp

```
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
```

```
<%
```

```
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String un=request.getParameter("un");
String em=request.getParameter("em");
String qu=request.getParameter("qu");
String an=request.getParameter("an");
Random r=new Random();
int fid=Math.abs(r.nextInt())/100000;
out.print("<center>");
out.println("<h1><u>Password Details</u></h1></center>");
out.print("<br>Your Uername = "+un);
out.print("<br>Your Email.Id = "+em);
out.print("<br>Secert Question = "+qu);
out.print("<br>Your Answer = "+an);
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into forgetpassword values("+fid+", '"+un+"', '"+
em+"', '"+qu+"', '"+an+"')";
```

```
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
```

readlogin.jsp

```
<%@ page import = "java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String un=request.getParameter("un");
String pw=request.getParameter("pw");
Random r=new Random();
int id=Math.abs(r.nextInt())/100000;
out.print("<center>");
out.println("<h1><u>Login Details</u></h1></center>");
```

```
out.print("<br>Your Username= "+id);
out.print("<br>Your Username= "+un);
out.print("<br>Your Password= "+pw);
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into login values("+id+", '"+un+"', '"+pw+"')";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}

}
%>
```

readmodifyattendance.jsp

```
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String id=request.getParameter("id");
int idumb=Integer.parseInt(id);
Random r=new Random();
int pid=Math.abs(r.nextInt())/100000;
out.print("<center>");
out.println("<h1><u>ID NUMBER</u></h1></center>");
out.print("<br>Id number = "+pid);
out.print("<br>Id number = "+id);
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into modifyattendance values("+pid+", "+id+")";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{

```

```
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
}%>
```

readmodifypatient.jsp

```
<%@ page import = "java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
```

```
<%
```

```
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String pn=request.getParameter("pn");
int pnumb=Integer.parseInt(pn);
Random r=new Random();
int cid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into modifypatient values("+cid+", "+pn+")";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
```



```
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
}%>
```

readmodifyward.jsp

```
<%@ page import ="java.sql.*, java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
```

```
<%
```

```
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String wn=request.getParameter("wn");
int wnumb =Integer.parseInt(wn);
Random r=new Random();
int Aid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
String sql="insert into modifyward values("+Aid+", "+wn+")";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
```

```
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
readupdatedoctor.jsp
<%@ page import = "java.sql.* , java.util.*" language="java" contentType="text/html;
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
Connection con=null;
Statement st=null;
String driver="com.mysql.jdbc.Driver";
String url="jdbc:mysql://localhost:3306/hospital";
String username="root";
String password="12345";
String did=request.getParameter("di");
Random r=new Random();
int cid=Math.abs(r.nextInt())/100000;
try
{
Class.forName(driver);
out.println("<p>Driver Registered");
con=DriverManager.getConnection(url,username,password);
out.println("<p>Connection Established");
st=con.createStatement();
out.println("<p>Statement Created");
```

```
String sql="insert into updatedoctor values("+cid+", "+did+")";
out.print("<p>Sql="+sql);
int x=st.executeUpdate(sql);
if(x==1)
{
out.print("<h1> Record Inserted Successfully</h1>");
}
else
{
out.print("<h1> Sorry! Record Could Not be Inserted</h1>");
}
}
catch(Exception e)
{
out.println("<p>Error Arises="+e.getMessage());
}
finally
{
try
{
if(st!= null)
st.close();
if(con!=null)
con.close();
}
catch(Exception e)
{
out.println("<p>Error Arises");
}
}
%>
```

INPUT/OUTPUT SCREEN

HOSPITAL MANAGEMENT SYSTEM

ENTER LOGIN DETAILS

Enter your username

Enter your password

HOSPITAL MANAGEMENT SYSTEM

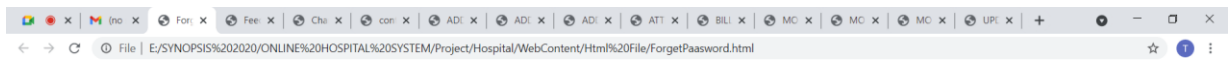
The screenshot shows a web browser window with a single tab titled 'Sign:'. The address bar displays the file path: `E:/SYNOPSIS%202020/ONLINE%20HOSPITAL%20SYSTEM/Project/Hospital/WebContent/Html%20File/Signup.html`. The main content area has a grey background and is titled 'ENTER YOUR DETIALS' in red. Below the title is a form with the following fields and options:

- User Name:
- Password:
- RetypePassword:
- First Name:
- Middle Name:
- Last Name:
- DATE OF BIRTH:
- Gender: ☐ Male ☐ Female ☐ Other
- Aadhar No.(Optional):
- Mobile No.:
- Email.Id:
- Secret Question:
- Answer:
- Address:
- Country Name:
- zip Code:

At the bottom of the form are two buttons: 'submit' and 'reset'.

The Windows taskbar at the bottom shows the search bar with the text 'Type here to search', several application icons, and the system clock displaying '03:15' and '30-05-2021'.

HOSPITAL MANAGEMENT SYSTEM



ENTER FORGETPASSWORD DETAILS

Enter your user name

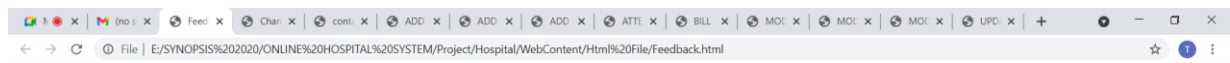
Email.Id

Secret Question

Answer



HOSPITAL MANAGEMENT SYSTEM



Enter Feedback Details

user name

EMAIL ID



HOSPITAL MANAGEMENT SYSTEM

File | E:/SYNOPSIS%202020/ONLINE%20HOSPITAL%20SYSTEM/Project/Hospital/WebContent/Html%20File/ChangePassword.html

WELCOME TO HOSPITAL

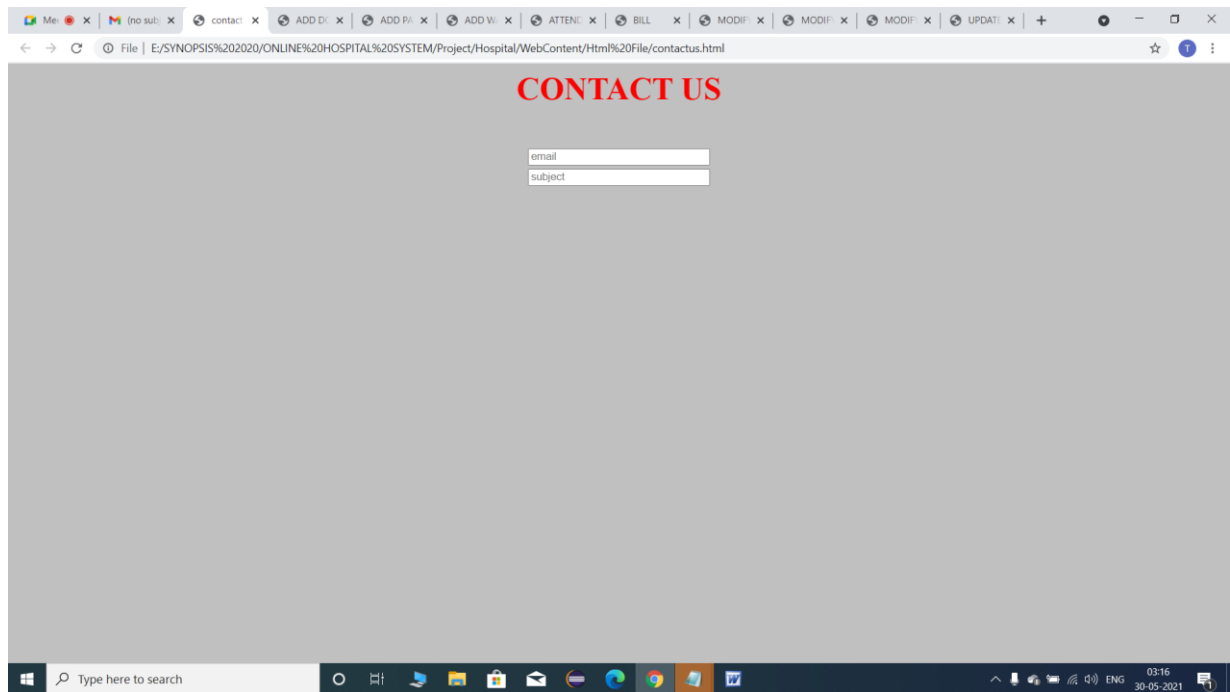
Enter your user name

Enter Current Password

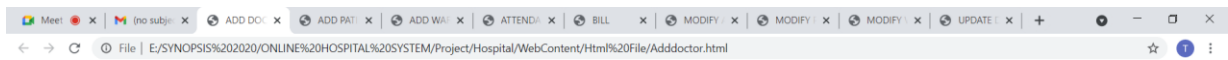
Enter New Password

Retype New Password

HOSPITAL MANAGEMENT SYSTEM



HOSPITAL MANAGEMENT SYSTEM

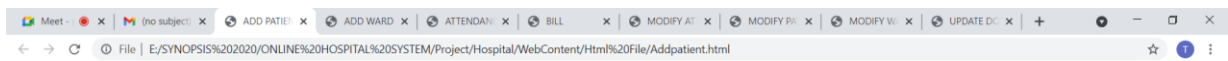


ENTER DOCTOR DETIALS

Doctor Name	<input type="text" value="doctormame"/>
Father name	<input type="text" value="fathername"/>
GENDER	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
DATE OF BIRTH	<input type="text" value="DD"/> <input type="text" value="MON"/> <input type="text" value="YYYY"/>
ADDRESS	<input type="text" value="address"/>
QUALIFICATION	<input type="text" value="qualification"/>
EXPERIANCE	<input type="text" value="experience"/>
LAST WORKED HOSPITAL	<input type="text" value="lastworkedhospital"/>
SALARY	<input type="text" value="salary"/>
	<input type="button" value="submit"/> <input type="button" value="reset"/>



HOSPITAL MANAGEMENT SYSTEM

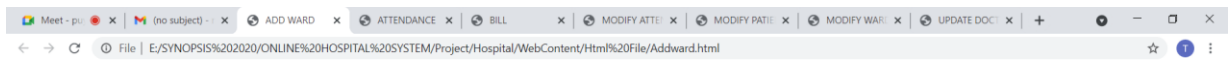


ENTER PATIENT DETAILS

Patient Name	<input type="text" value="patientname"/>
Age	<input type="text" value="age"/>
Gender	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Address	<input type="text" value="Address1"/>
Consult doctor	<input type="text" value="consultdoctor"/>
Problem	<input type="text" value="problem"/>
Admitted	<input type="text" value="Admitted"/>
Non admitted	<input type="text" value="nonadmitted"/>
Fee	<input type="text" value="fee"/>
Diagnosis	<input type="text" value="diagnosis"/>
Mobile No.	<input type="text" value="mobile no"/>
Contact number	<input type="text" value="contactnumber"/>
Email.Id	<input type="text" value="email id"/>
	<input type="button" value="submit"/> <input type="button" value="reset"/>



HOSPITAL MANAGEMENT SYSTEM



ADD WARD DETAILS

WARD NAME

WARD TYPE

WARD MODE

TOTAL BED

COST

ROOM TYPE ☐ AC Room
☐ Non-ac Room



HOSPITAL MANAGEMENT SYSTEM

Meet - puy... x | (no subject) - raf... x | ATTENDANCE x | BILL x | MODIFY ATTEND... x | MODIFY PATIENT x | MODIFY WARD x | UPDATE DOCTOR x | +

File | E:\SYNOPSIS\2020\ONLINE\HOSPITAL\SYSTEM\Project\Hospital\WebContent\Html\Attendance.html

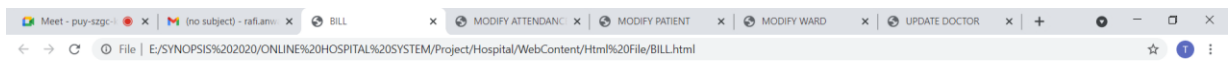
ENTER ATTENDANCE

Date Of Attendance	DD ▾ MON ▾ YYYY ▾
Incoming time	<input type="text" value="incomingtime"/>
Outgoing time	<input type="text" value="outgoingtime"/>
Thank involved	<input type="text" value="thankinvolved"/>
	<input type="button" value="submit"/> <input type="button" value="reset"/>

Type here to search

03:17 30-05-2021

HOSPITAL MANAGEMENT SYSTEM

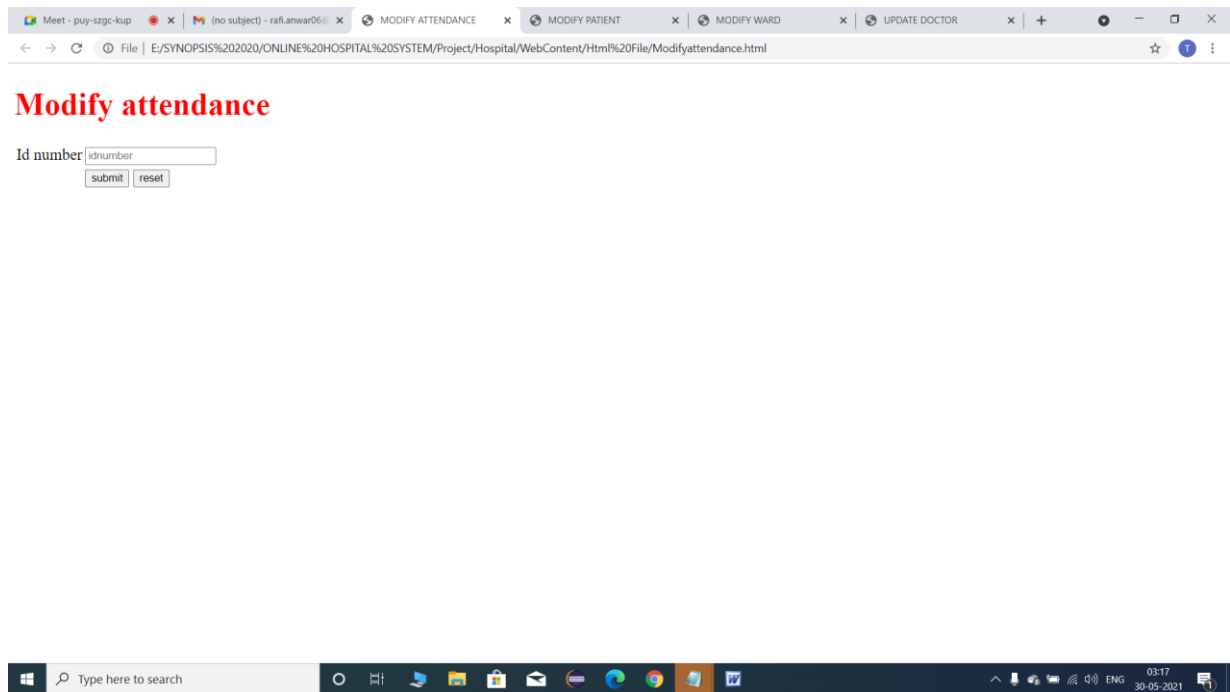


ENTER BILL

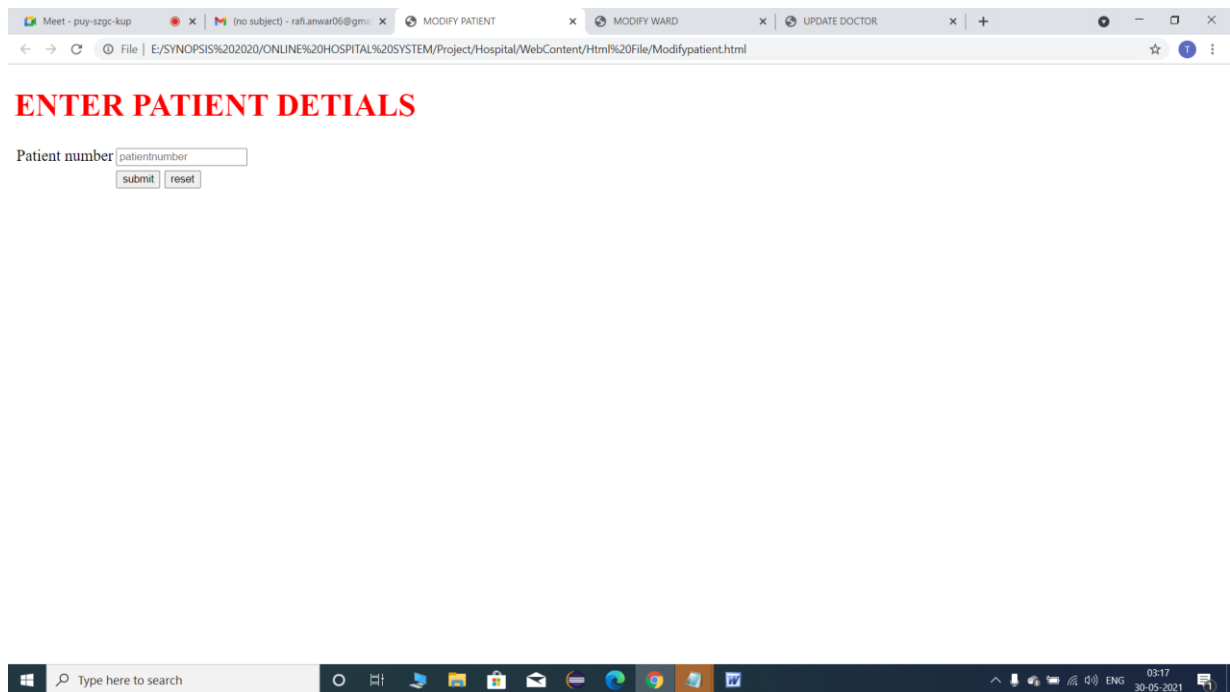
Patient Id number	<input type="text" value="patientidnumber"/>
Patient name	<input type="text" value="patientname"/>
Consult doctor	<input type="text" value="consultdoctor"/>
Diagnosis	<input type="text" value="dianosis"/>
CONTACT NUMBER	<input type="text" value="contactnumber"/>
Mobile number	<input type="text" value="mobilenumber"/>
Email.Id	<input type="text" value="email.id"/>
Address	<input type="text" value="address"/>
	<input type="button" value="submit"/> <input type="button" value="reset"/>



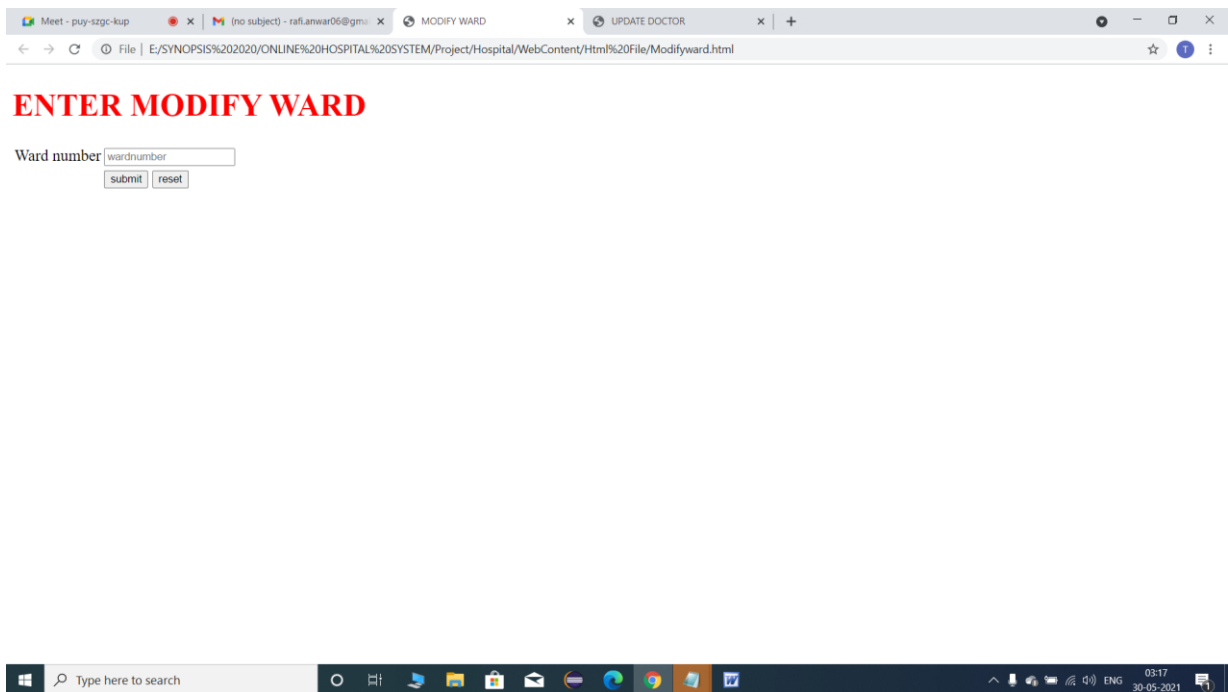
HOSPITAL MANAGEMENT SYSTEM



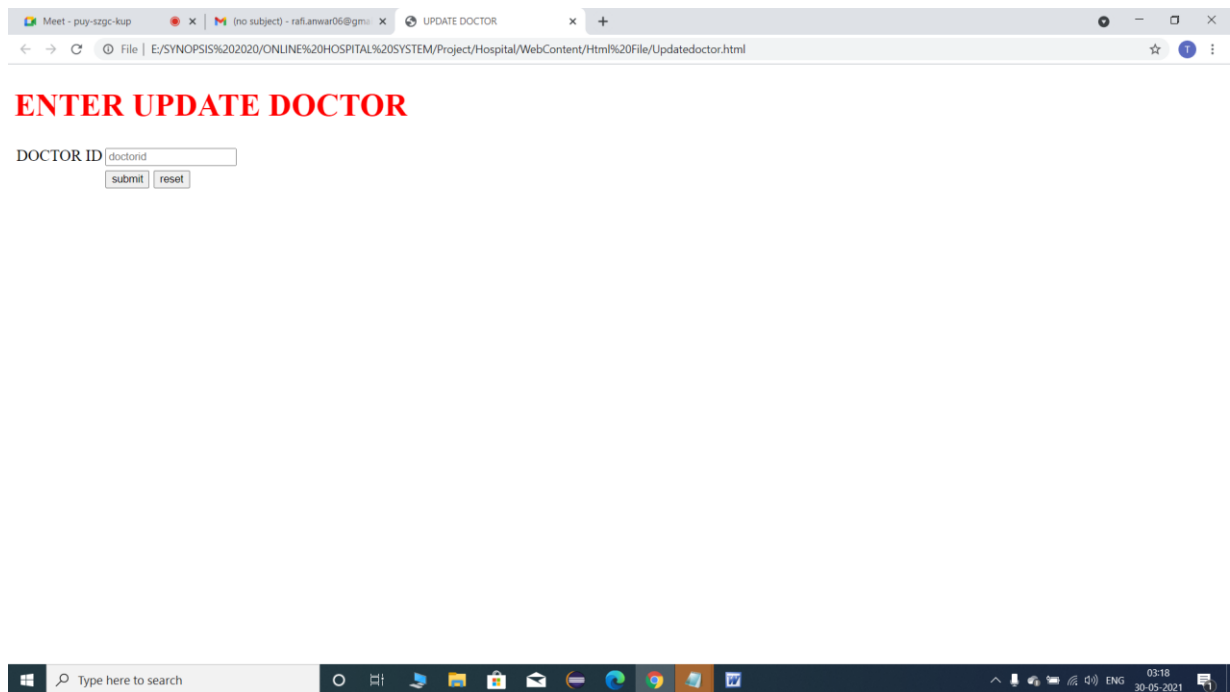
HOSPITAL MANAGEMENT SYSTEM



HOSPITAL MANAGEMENT SYSTEM



HOSPITAL MANAGEMENT SYSTEM



TESTING

TESTING

Once source has been generated, software must be tested to uncover and correct as many errors as possible before delivery to the customer.

The test plan is an important product of software design. A test can pre describe various kinds of activities that will be performed to demonstrate that the software product is working as required.

Any engineered test can be tested on one of the following two ways: -

1. WHITE BOX TESTING
2. BLACK BOX TESTING

White box Testing

White box testing is also called glass box testing. It is a test case design that uses control structure of the procedure design to derive test cases. Using white box testing aids, the software engineer can derive test cases that:

1. Guarantee all independent paths within a module have been exercised at least once using mechanism flow graph notation.

2. Exercise all logical decision on their trace and false sides. This is called conditional testing.
3. Execute all loops at their boundaries and within their operational bounds. This is called loop testing.
4. Exercise all internal data structure to ensure their validity. This is called "Data flow testing".

Basis Path Testing: -

Basis path testing is a white-box testing technique first proposed by *Tom McCabe*. The basis path method enables the test case to derive a logical complexity measure of a procedural design and this measure as a guide for defining a basis set of execution paths. Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least one time during testing.

Condition Testing: -

Condition testing is a test case design method that exercise the logical conditions contained in a program module. The condition testing method focuses on testing each condition in the program. Condition testing strategies have two advantages, first, measurement of test coverage of a condition is simple, second, and the test coverage of conditions in a program provides guidance for the generation of additional tests for the program.

Purpose of condition testing is to detect not only errors in the conditions of a program but also other errors in the program. *Branch testing* is simplest condition testing strategy.

Data Flow Testing: -

The data flow testing method selects test paths of a program according to the locations of definitions and uses of variables in the program. Data flow testing strategies are useful for selecting test paths of a program containing nested *if* and *loop* statements.

Loop Testing: -

It is also a white-box testing technique that focuses exclusively on the validity of loop constructs. Four different classes of loops can be defined: simple loops, concatenated loops, nested loops, and unstructured loops.

Black box Testing

Black box testing is also called behavioral testing, which focuses on the functional requirements of the software. Black box testing enables the software engineer to derive sets of input condition that will fully exercise the functional requirement for the program.

Black box testing attempts to find error in the following categories:

1. Incorrect or missing function.

2. Interface error.
3. Error in data structure or external database access.
4. Behavior or performance error.
5. Initialization or termination error.

Unlike white-box testing, which is performed early in the testing process, Black box testing tends to be applied during later stages of testing. By applying Black-box techniques, we derive a set of test cases that satisfy the following criteria:

- (1) Test cases that reduce, by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing and
- (2) Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specific test at hand.

Graph-Based Testing Methods: -

The first step in black-box testing is to understand the objects that are modeled in software and the relationships that connect these objects. Once this need has been accomplished, the next step is to define a series of tests that verify, "All objects have the expected relationship to one another."

Equivalence Partitioning: -

Equivalence partitioning is a black-box testing method that divides the input domain of a program into classes of data from which test cases can be derived. An ideal test case single-handedly uncovers a class of errors that might otherwise require many cases to be executed before the general error is observed. Equivalence partitioning strives to define a test case that uncovers classes of error, thereby reducing the total number of test cases that must be developed. Test case design for equivalence partitioning is based on an evaluation of equivalence classes for an input condition.

Boundary Value Analysis: -

Boundary value analysis is a test case design technique that complements equivalence partitioning. Rather than selecting any element of an equivalence class, BVA leads to the selection of test cases at the “edges” of the class. By using these testing methods in the present software many defects and errors present has been removed and it looks a good software with no or minimum errors.

TESTING STRATEGY

The strategy provides a road map that describes the steps to be conducted as part of testing, when these steps are planned and then undertaken, and how much effort, time, and resources will be required.

A number of software testing strategies have been proposed. All provides the software developer with a template for testing and all have the following generic characteristics:

- ☞ Testing begins at the component level and works “outward” toward the Integration of the entire computer-based system.
- ☞ Different testing techniques are appropriate at different points in time.
- ☞ Testing is conducted by the developer of the software and the independent test Group.
- ☞ Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

Unit Testing: -

Unit testing focuses verification efforts on the smallest unit of software design-the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested

to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structure are exercised to ensure that all statements in a module have been executed at least once. And finally, all errors handling paths are tested.

Unit testing can be performed from the bottom up, starting with smallest and lowest-level modules and proceeding one at a time. For each module in bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embed within the larger system.

Integration Testing: -

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and built a program structure that has been dictated by design.

There are two types of integration testing

- (a) *Top-down Integration* and
- (b) *Bottom-up Integration*.

Validation Testing: -

Software validation is achieved through a series of block-box tests that demonstrate conformity with requirements. A test plan outlines the classes

of tests to be conducted and a test procedure defines specific test cases that will be demonstrated conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied, all behavioral characteristics are achieved, all performance requirements are attained and human-engineered and other requirements are met.

System Testing: -

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. System testing makes a logical assumption that if all the parts of the parts of the system are correct, the goal will be successfully activated. Another reason for system testing is its utility as a user-oriented vehicle before implementation.

System testing consists of following steps:

- Program testing
- String testing
- Recovery testing
- Security testing
- Stress testing

- Performance testing

These given testing strategies have been used for the present project software. It took a lot of time to test every possible condition for the system and the software can be said "*Tested OK*" up to a reasonable limit

DEBUGGING

Software testing is a process that can be systematically planned and specified. Test case design can be conducted, strategy can be defined, and result can be evaluated against prescribed expectation.

Debugging occurs as a consequence of successful testing. That is when a test case uncovers an error, debugging is the process that results in the removal of the error. Although debugging can & should be an orderly process, it is still very much an art. A software engineers, evaluating the result of a test, is often confronted with a “symptomatic” indication of a software problem. That is, the external manifestation of the error and the internal cause of the error may have no obvious relationship to one another. The poorly understood mental process that connects a symptom to a cause is debugging.

The debugging process

Debugging is not testing but always occurs as a consequence of testing the debugging process begins with the execution of a test case. Results are assessed and a lack of correspondence between expected and actual performance is encountered. In many ways cases, the no corresponding data are a symptom of an underlying cause as yet hidden. The debugging process attempts to match symptom with cause, thereby leading to error correction.

The debugging process will always have one of two outcomes. (1) The cause will be found and corrected, or (2) the cause will not be found. In the latter case, the person performing debugging may suspect a cause, design a test case to help validate that suspicion, and work toward error correction in the iterative fashion.

Why is debugging so difficult? In all likelihood, human psychology has more to do with an answer than software technology. However, a few characteristics of bugs provide some clues.

1. The symptom and the cause may be geographically remote. That is the symptom may appear in one part of program, while the cause may actually be located at a site that is far removed. Highly coupled program structures exacerbate this situation.
2. The symptom may disappear (temporarily) when another error is corrected.
3. The symptom may actually be caused by non-error (e.g., round-off inaccuracies).
4. The symptom may cause by human error that is not easily traced.
5. The symptom may be a result of timing problems, rather than processing problems.
6. It may be difficult to accurately reproduce input condition (e.g. a real time application in which input ordering is indeterminate).

7. The symptom may be intermittent. This is particularly common in embedded systems that couple h/w & s/w inextricably.
8. The symptom may be due to causes that are distributed across a number of tasks running on different processors.

During debugging we encounter error that range from mildly annoying (e.g. an incorrect output format) to catastrophic (e.g. the system fails, causing serious economic or physical damage). As the consequences of an error increase, the amount of pressure to find the cause also increases. Often, pressure sometimes forces software developed to fix one error and at the same time introduce two more.

Psychological considerations

Unfortunately, there appears to be some evidence that debugging prowess is an innate human trait. Some people are good at it and other isn't.

Although experimental evidence on debugging is open to many interpretations, large variances in debugging ability have been reported for programmers with the same education and experience.

Debugging is one of the more frustrating parts of programming. It has elements of problem solving or brainteasers, coupled with the annoying recognition that you have made a mistake. Heightened anxiety and the unwillingness to accept the possibility of error increase the task difficulty. Fortunately, there is a great sign of relief and a lessening of tension when the bug is ultimately corrected.

Although it may be difficult to “learn” debugging, a number of approaches to the problem can be proposed.

Debugging Approaches

Regardless of the approach that is taken, debugging has one overriding objective: to find and correct the cause of an s/w error. The objective is realized by a combination of systematic evaluation, intuition, and luck.

Bradley describes the debugging approach in this way.

CODE EFFICIENCY

The code is called efficient when it is easy to understand, test and maintain. The code written in this application software (pharmaceutical distribution system) is easy to understand, tested and maintained. Each module is separately coded so that it would be easy to understand by other programmer. It is less error prone. It can be easily maintained.

The coding of each form is done separately and I think any programmer won't get any sort of problem during implementing this project and so far I am concerned with this coding it is very user friendly. Each button command is user friendly defined in this project so that error can be recovered and module wise coding will ease the programmer from every corner.

Every care has been taken to trap the most common error message to make the code efficient. A code walk-through is an informal technique for analysis of the code. A code walk-through of a module is undertaken after the coding of the module is complete. In this technique, after a module has been coded, members of the development team select some test cases and simulate execution of the code by hand making the coding of the module efficient.

Code used in the Application is efficient by using following features:

- ☞ Object oriented approach (Class, Inheritance, Reusability)
- ☞ Platform Independence:

The compiler generates an architecture-neutral object file format—the compiled code is executable on many processors, given the presence of the Java run time system. The Java compiler does this by generating byte code instructions, which have nothing to do with particular computer architecture. Rather, they are designed to be both easy to interpret on any machine and easily translated into native machine code on the fly.

☞ High Performance

While the performance of interpreted byte codes is usually more than adequate, there are situations where higher performance is required. The byte codes can be translated on the fly (at run time) into machine code for the particular CPU the application is running on.

Java is interpreted, so it is too slow for serious applications. But Speed is not everything! Snazzy feature do not count.

Business drives many decisions, and here business sides of affairs to the technology contort. Choosing a P/L to execute a Project because the language is technically most suited, may not make business sense.

Unfortunately, decisions are almost always driven by reasons outside technology.

Let us take these facts.

- ✚ Chances of bagging a Project
- ✚ Chances of finding expertise
- ✚ Vendor support

✚ Availability of a Future Road map

✚ The Benchmark and scorecard

Language	Availability	Final Score
C/C++	4,000,000	22
Java	3,000,000	25
VB.Net	3,000,000	23
ASP.Net	1,000,000	19
C#	2,000,000	24
Delphi	800,000	20
PHP	400,000	16

Final Conclusion: Java and C# are the clear leaders followed by C, C++, Delphi and Visual Basic.Net, when it comes to the choice for s/w development.

OPTIMIZATION OF CODE

Optimization means to make as perfect or effective as possible. The software designer should be concerned with developing a representation of software that will meet all functional and performance requirements. Design optimization should strive for the smallest number of modules that is consistent with effective modularity and least complex data structure that adequately serves information requirement.

As the optimization is being used for the application developed, the following points are being considered.

- Coding is done in appropriate programming language.
- User can easily navigate from one module to another.
- The Reports can be generated easily by selecting the required menu item.

ERROR HANDLING

Error message and warnings are “bad news” delivered to users of interactive systems when something has gone awry. At their worst, error message and warnings impart useless or misleading information and serve only to increase user frustration. There are few computer users who have not encountered an error of the form: *“Application XXX has been forced to quit because an error of type 1023 has been encountered.”* Somewhere an explanation for error 1023 must exist; otherwise, why would the designers have added the identification? Yet, the error message provides no real indication of what went wrong or where to look to get additional information. An error message presented in this manner does nothing to assuage user anxiety or to help correct the problem.

In general, every error message or warning produced by an interactive system should have the following characteristics.

- The message should describe the problem in language the user can understand.
- The message should provide constructive advice for recovering from the error.
- The message should indicate any negative consequence of the error (e.g. potentially corrupted data files) so that the user can check to ensure that they have not occurred (or correct them if they have).
- The message should be accompanied by an audible or visual cue. That is a beep might be generated to accompany the display of the message, or the message might flash momentarily or be displayed in a color that is easily recognizable as the "error color".
- The message should be nonjudgmental. That is the wording should never place blame on the user.

Because no one really like bad news, few users will like an error message no matter how well designed. But an effective error message philosophy can do much to improve the quality of an interactive system and will significantly reduce user frustration when problem do occur.

VALIDATION CHECKS

The work products produced as consequences of requirements engineering are assessed for quality during a *validation* step. Requirement validation examines the specification to ensure that all software requirements have been stated unambiguously; that inconsistencies, omissions, and errors have been detected and corrected; and that the work product conforms to the standards established for the process, the project, and the product. The primary requirements validation mechanism is the formal technique review. The review team that validates requirements includes s/w engineering, customer, users, and other stakeholders who examine the specification looking for error in content or interpretation, areas where clarification may be required, missing information, inconsistencies (a major problem when large products or systems are engineered), conflicting requirements, or unrealistic (unachievable) requirements.

COST ESTIMATION OF THE PROJECT

Software cost and effort estimation will never be an exact science as Bennatan reports that 50 percent of software developers continue to struggle with estimation and that software size and development time are very difficult to estimate accurately. Too many variables like human, technical, environmental, political can affect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to a series of systematic steps that provide estimates with acceptable risk. To achieve reliable cost and effort estimates, a number of options arise:

1. Delay estimation until late in the project.
2. Base estimates on similar projects that have already been completed.
3. Use relatively simple decomposition techniques to generate project cost and effort estimates.
4. Use one or more empirical models for software cost and effort estimation.

There are so many approaches for the estimation of cost and efforts. It may be one of the following: -

1. Problem-Based estimation
2. LOC-Based estimation
3. FP-Based estimation

I am presenting here **FP-Based estimation** for this project. This estimation goes through a no. of steps which has been described below.

Measurement Parameter	Count (P_i)	Weighting Factor			Total (WC_i)
		Simple	Average	Complex	
No. of user inputs	8	3 $8 \times 3 = 24$	4 $8 \times 4 = 32$	5 $8 \times 5 = 40$	96
No. of user ouputs	4	4 $4 \times 4 = 16$	6 $4 \times 6 = 24$	7 $4 \times 7 = 28$	68
No. of user inquiries	7	3 $7 \times 3 = 21$	5 $7 \times 5 = 35$	8 $7 \times 8 = 56$	112
No.of Internal Logical Files	7	2 $7 \times 2 = 14$	4 $7 \times 4 = 28$	6 $7 \times 6 = 42$	84
No.of External Logical Files	3	4 $3 \times 4 = 12$	5 $3 \times 5 = 15$	6 $3 \times 6 = 18$	45
Count Total (UFPC)					405

GAC (General Application Characteristics) List : -

Sl. No.	GAC	DI (F _i)
1.	Data Communication	5
2.	Distributed Functions	4
3.	Criticality of Performance	1
4.	Configuration of the system	3
5.	Transaction rates	3
6.	On-line data entry	4
7.	Design for end-user efficiency	4
8.	Online updation	4
9.	Complexity	2
10.	Reusability or usability in other applications	5
11.	Ease of installation	4
12.	Operational ease	4
13.	Multiple sites	1
14.	Facilitate change	4
TDI (Total Degree of Influence)		48

$$\begin{aligned}\text{VAF (Value Adjustment Factor)} &= (\text{TDI} \times 0.01) + 0.65 \\ &= (48 \times 0.01) + 0.65 &= 1.13\end{aligned}$$

$$\begin{aligned}\text{FPC (Function Point Count)} &= \text{UFPC} \times \text{VAF} \\ &= 405 \times 1.13 \\ &= 457.65\end{aligned}$$

$$\text{FPC} = 457$$

If E is the effort in person-months or person-years then by Albrecht and Gaffney model :

$$\begin{aligned}E &= -91.4 + 0.3555 \text{ FP} \\ &= -91.4 + (0.355 \times 457) \\ &= -91.4 + 162.24 \\ &= 70.84\end{aligned}$$

$$E = 70 \text{ person - months}$$

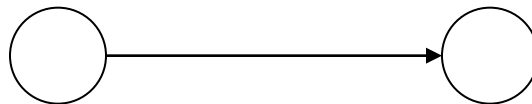
Here estimating cost is irrelevant as this project being developed for submission as final project by me.

PLANNING & SCHEDULING

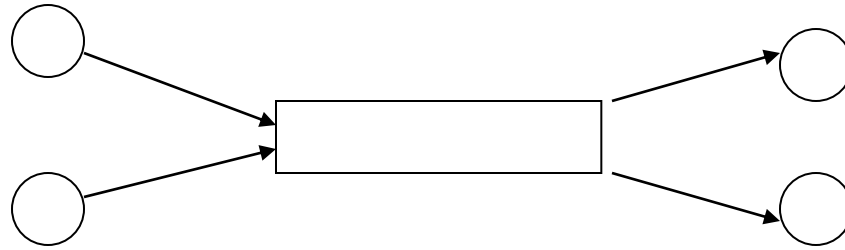
PERT CHART

PERT stands for “program evaluation and review technique” it is used extensively for project planning and control. As a result, the most powerful project planning and management tools are now available to people oriented business. For more than the technical faults, it was found that PERT provided a focus around which managers could brainstorm and put their ideas together. It proves to be great communication medium by which thinkers and planners at level could communicate their ideas, their doubts and fears to another level. It is a diagram in which all activities are shown as a network of dependent activities. It shown:

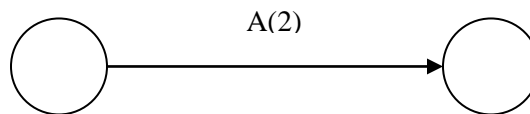
4. Activities: it is represented as an arrow. The tail of an arrow indicates the start of an activity and its head represents the
5. Completion. The start and end events of the activities are shown as circle.



6. Milestones: Milestone marks the end of significant phase. Several activities can start or end at milestones. It is represented as rectangle.



Duration of each task: we show duration of each task by labeling between start and end event of each activity. It is shown that how much time has spent in that activity.



The most important aspect of a PERT diagram is the critical path. A critical path is a chain of critical task. The critical path comprises of critical task, which cannot take longer than estimated without delaying the entire project. The path of activities with zero floats

From the start of the project to the end of the project is called the critical path. A critical path is the longest path on the network.

We passed through such system activities in period of project completion. So, here we identified them as activities and time spend for it. Such system activities are:

.

HOSPITAL MANAGEMENT SYSTEM

System activities:	Time (in days)
L. Project definition	2
M. System analysis	3
N. Feasibility study	2
O. Software engineering	5
P. Software/ hardware requirements	2
Q. System design	10
R. Coding	20
S. Validation check	4
T. Testing	4
U. Cost estimation	3
V. User acceptance	5

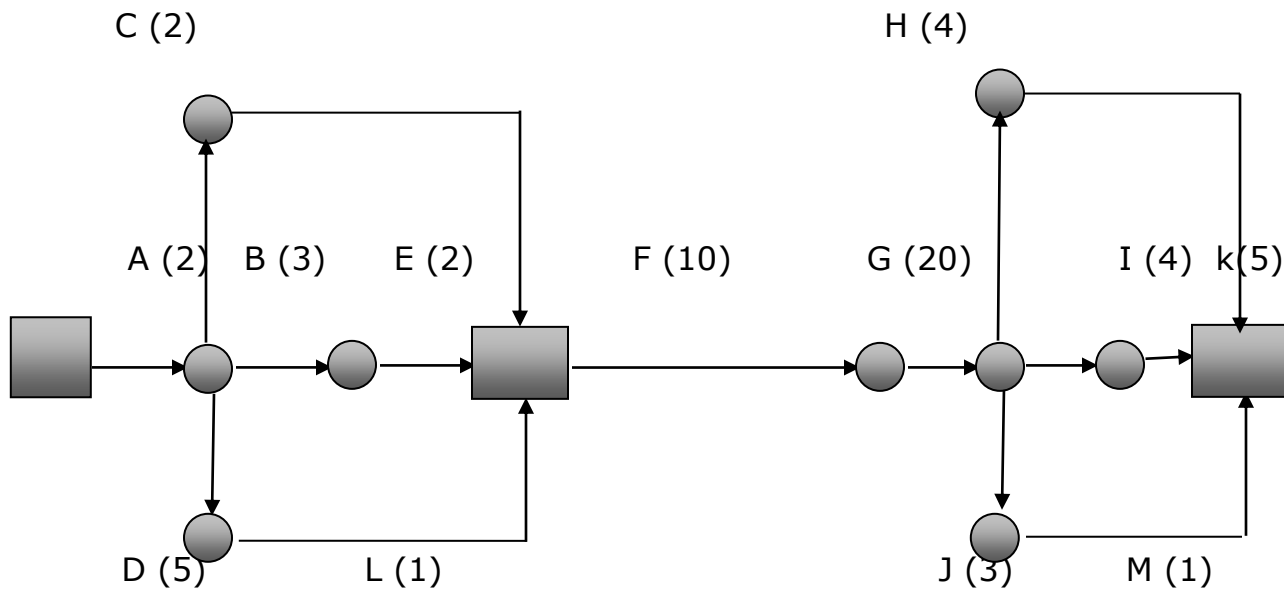
GANTT CHART

It is useful tool for scheduling project activities and tracking is Gantt chart. It is a basic means of presenting a project program visually. A Gantt chart is a bar chart and it gets its name from Harvey Gantt, a pioneer of operation research. Today is largely used to presents the distinct stages of a project. A Gantt chart is composed of vertical axes (usually in day increments) that is, it helps me to plots tasks or resources against time.

We defined vertical axis as system activity and horizontal axis as days. Each system activity is required some time, so, we drawn bar against days for activity. Here also defined time spend in activity. Such as:

<u>System activities:</u>	<u>Time (in days)</u>
J. Project definition	2
K. System Analysis	3
L. Feasibility study	2
M. Software engineering	5
N. Software/Hardware requirements	2
O. System design	10
P. Coding	20
Q. Validation check	4
R. Testing	4

HOSPITAL MANAGEMENT SYSTEM



Here , the critical path is, A-> B-> E->F->G->I->K= 46 Days

SECURITY

Security measures are one of the best efforts to build any information system well secure. The main aim of security measures is to verify that protection mechanism built into a system to protect it from improper penetration.

Security encompasses a set of measures taken to guard against theft, attack, crime and espionage or sabotage. Security implies the quality or state of being secure that is a relief from exposure to danger and acting so as to make safe against adverse contingencies.

Keeping all these views in the mind I also implement a high level of security measures; the user can only use the software after entering the correct password, the password is case sensitive and does not except null entries, if the user enter wrong password he always prompt for reentering the user name and password, and I also provide a feature to change the user name and password options which is used in case of leakage of password, which if happened may cause to break the security measures.

Database / data Security: -

Database security has been imposed in my project by login process. Here the administrator has been given complete authority to access and handle the database in the manner he wants, whereas the general user has been restricted to some specific works over the database.

Creation of User profiles and access rights : -

Only the administrator has been given to create new user and to define access rights for them.

IMPLEMENTATION

System Implementation is the construction of the new system and delivery of that system into production that means day-to-day operation.

Implementation is the process of having systems personal check out and put new equipment into use, train users, installs the new application, and constructs any files of data needed to use it. Depending on the sizes of the organization that will be involved in using the application and the kiosk associated with its uses, system developers may choose to test the operation in only one area of the firm. Sometimes this will run the old and new system together to compare the results. Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new; replacing an existing manual or automated system, or it may be major modification into an existing system. Since organization systems should kept pace, in this sense we can say that Implementation is an ongoing process and it is also said that Implementation is the process of bringing system in real world application. Here I have built the system which serve the purpose of **HOSPITAL MANAGEMENT SYSTEM** which in future they can implement their system without any problem.

EVALUATION

Evaluation of the system is performed to identify its strengths and weaknesses. Evaluation measures the system's performance against pre-defined requirements. It determines how well the system continues to meet performance specifications. It also provides information to determine whether major re-design or modification is required.

A post-implementation review is an evaluation of a system in terms of the extent to which the system accomplishes stated objectives and actual project costs exceed initial estimates. It is usually a review of major problems that need converting and those that surfaced during the implementation phase.

The actual evaluation can occur along any of the following dimension, which I have performed in my system.

From Operational point of view there is Assessment of the manner in which the system function, including ease of use, response time, suitability of information format's overall reliability, and level of utilization are performed.

From Organizational point of view there is Identification and measurement of benefits to the organization in such areas as financial concerns (cost, revenue, and profit), operational efficiency, and competitive impact.

Includes impact on internal and external information flows are performed.

From User Manager Assessment point of view there is Evaluation of the attitudes between staff and manager within the organization are performed.

The present software has been also evaluated for knowing its actual working in the practical situation. During evaluation period the functioning of the software has been found good.

MAINTENANCE

System Maintenance phase is the last phase of system development life cycle. System maintenance is actually the implementation of the post-implementation review plan.

Maintenance covers a wide range of activities including correcting coding and design errors, updating documentation and test data and upgrading user support. Many activities classified as maintenance actually fall under enhancements. Maintenance means restoring something to its original position. Unlike hardware, software does not wear out; it is corrected. In contrast, enhancement means adding, modifying or redeveloping the code to support changes in the specifications. It is to keep with changing user needs and the operational environment.

The keys to reduce the need for maintenance while making it possible to carry on with essential tasks more efficiently are as follows:

- More accurately defining the user's requirement during systems development.
- Preparation of system documentation in a better way.
- Using more effective ways for designing processing logic and communicating it to project team members.
- Making better use of existing tools and techniques.

So, finally we can say that this includes program maintenance and system improvements.

LIMITATION OF PROJECT

- This Project doesn't cover all schemes that are issued by institute.
- It doesn't support network.
- It doesn't make any type of backup file.
- Not any sorting technique used for fast access.
- It supports sequential searching technique only.
- Only keyboard & mouse used as input device.
- Although there may be many other works in HOSPITALS but this project deserves only girls related information.
- There are no any printing techniques are used to get the hardy.
- Some terms, which are used in project, is assumed.

FUTURE SCOPE OF THE PROJECT

In future, this project support networking through which it may be connected to the Internet. Through these facilities agency can consult with the showroom and get useful instruction.

- In future it supports multi-user operating system like UNIX.
- In future this project support distributed database rather than centralized database.
- Storing huge amount of data for further use.
- Reducing manual efforts for maintaining the system.
- Reducing the lead-time.
- It gives the correct information about the data, which is further interrupted.
- For better support, the users this project also support the following technique in future.
 - Backup file
 - Fast searching technique
 - Queries
 - Support broad range input and output device.

BIBLIOGRAPHY

BOOK NAME

AUTHOR NAME

The Complete References of Java

Herbert Schildt

Software Engineering

Roger S.Pressman

SQL, PL/SQL

Ivan Bay Ross

System Analysis & Design

Siddrath

An Introduction to DBMS

Bipin C.Desai

IGNOU Course material