Hospital Management System

Submitted in partial fulfilment of the Requirements for the Degree of

Master of Computer Applications

A PROJECT REPORT

Submitted by

PRACHI SINGH

(University Roll No 1900290140023)

PRIYANKA PATEL

(University Roll No 1900290140025)

Batch:2019-2022

Under the Supervision of

S.D Mishra

(Assistant Professor)

DEPARTMENT OF COMPUTER APPLICATIONS

DR.APJ ABDUL KALAM TECHNICAL UNIVERSITY

LUCKNOW

(Formerly Uttar Pradesh Technical University, Lucknow)

DECLARATION

I hereby declare that the work presented in this report entitled “hospital management system", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name :

Roll. No. :

Branch :

(Candidate Signature)

CERTIFICATE

Certified that **Prachi Singh (University Roll No 1900290140023), Priyanka Patel (University Roll No 1900290140025)**, have carried out the project work having "Hospital Management System" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU),Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Prachi Singh
University Roll No. 1900290140023
Priyanka Patel
University Roll No. 1900290140025

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

S.D Mishra
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava

Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multi specialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

ACKNOWLEDGEMENTS

Success in life is never attained single handed. My deepest gratitude goes to my thesis supervisor, **S.D Mishra** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Prachi Singh

Priyanka Patel

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multi specialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and

activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

1.2 PROJECT SCOPE

When patients arrive they make an appointment at the reception to consult a Doctor. These are being recorded in a file. Then again the patients diagnosed symptoms related disease details, ward details and other necessary details are being recorded and those files are being stored in special locations. Calculation of bills and inventory are done manually. As the current system is a file based one, management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also could be misplaced by losing data and information. Limited storage space of the files is another issue that they currently face when the management is manually done. There occurs an issue with the organization of data information and schedules and running the process methodically which leads to the manual system malfunctioning. Hospital Management System . If we want to check a previous record of a patient or other detail. Management will be in a great problem. It’s a tough and time taking process to search for a record in a file. Keeping files takes much time and waste much precious man hours. The tendency of making mistakes is high when functioning manually. It is hard to relay on the accuracy of calculations done manually too. It is more obvious for problems to arise. We plan to overcome the above mentioned problems through a standalone application, to manage the major functions of the Hospital System. The hospital management system we are going to implement will be covering all basic processes done in the hospital. It would handle Employee and Salary management, Patient, Theatre and ward Management, Laboratory management, Transport Management, Pharmacy Management, OPD management and emergency management.

In OPD unit, with the OPD and Consultation Management system, the manual doctors channeling details entering process has automated. So the staff does not need to spend time on writing appointment records and updating them in files. And the number issuing process becomes easier and efficient. And keeping the track of patients and medical prescription details allow them to review the details whenever needed. Implementing the Employee & Salary Management system we record Attendance, shifting of employees, their holidays and consulting doctors' schedules.

1.3 REQUIREMENTS

FUNCTIONAL REQUIREMENT

Registration Process of SRS (Software Requirements Specification)

● Adding Patients: The Hospital Management enables the staff in the front desk to include new patients to the system.

● Assigning an ID to the patients: The HMS enables the staff in the front desk to provide a unique ID for each patient and then add them to the record sheet of the patient. The patients can utilize the ID throughout their hospital stay.

Check Out of SRS:

● Deleting Patient ID: The staff in the administration section of the ward can delete the patient ID from the system when the patient's checkout from the hospital.

● Adding to beds available list: The Staff in the administration section of the ward can put the bed empty in the list of beds-available.

Report Generation of SRS:

● Information of the Patient: The Hospital Management System generates a report on every patient regarding various information like patients name, Phone number, bed number, the doctor's name whom its assigns, ward name, and more.

● Availability of the Bed: The Hospital Management system also helps in generating reports on the availability of the bed regarding the information like bed number unoccupied or occupied, ward name, and more.

Database of SRS:

● Mandatory Patient Information: Every patient has some necessary data like phone number, their first and last name, personal health number, postal code, country, address, city, 'patient's ID number, etc.

● Updating information of the Patient: The hospital management system enables users to update the information of the patient as described in the mandatory information included.

Non Functional Requirements

There are a lot of software requirements specifications included in the non-functional requirements of the Hospital Management System, which contains various process, namely Security, Performance, Maintainability, and Reliability.

Security:

● Patient Identification: The system needs the patient to recognize herself or himself using the phone.

● Logon ID: Any users who make use of the system need to hold a Logon ID and password.

● Modifications: Any modifications like insert, delete, update, etc. for the database can be synchronized quickly and executed only by the ward administrator.

● Front Desk Staff Rights: The staff in the front desk can view any data in the Hospital Management system, add new patients record to the HMS but they don't have any rights alter any data in it.

● Administrator rights: The administrator can view as well as alter any information in the Hospital Management System.

Performance:

● Response Time: The system provides acknowledgment in just one second once the 'patient's information is checked.

● Capacity: The system needs to support at least 1000 people at once.

● User-Interface: The user interface acknowledges within five seconds.

● Conformity: The system needs to ensure that the guidelines of the Microsoft accessibilities are followed.

Maintainability:

● Back-Up: The system offers the efficiency for data back up.

● Errors: The system will track every mistake as well as keep a log of it.

Reliability:

● Availability: The system is available all the time.

CHAPTER 2

LITERATURE REVIEW

Together with the rapidly increasing popularity of the Internet in recent years, there is an increasing demand for methodologies and technologies for e-learning. E-learning is an interactive learning in which the learning content is available on-line and provides

Automatic feedback to the student's learning activities.

It is an online platform where teachers and students can collaborate in order to improve student achievement. For students ,it allows them to access their class online and work in it.

Deep customization and flexibility lets us do exactly what we want. Data –Based insights allo us to understand and improve the e learning process.

Unrivalled security for our data ,media and users.constant ongoing improvement scope.

It reduce paper work.

CHAPTER 3

FEASIBILITY STUDY

Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

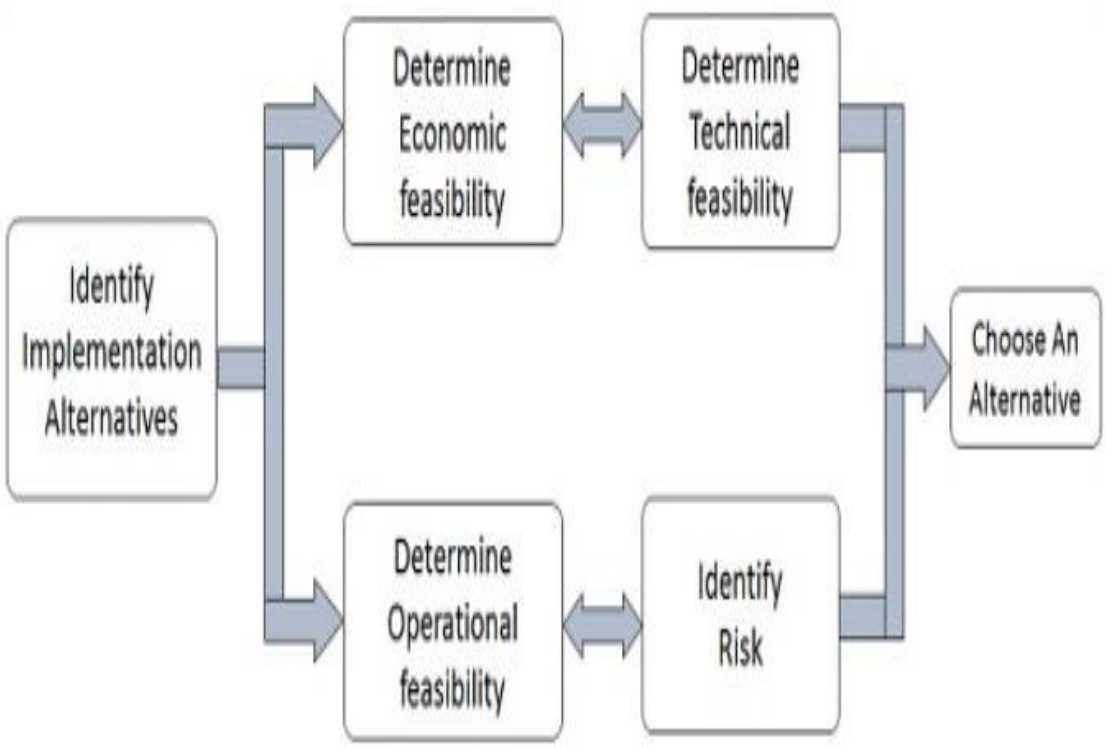
- It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
- It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
- The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
- The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

Steps Involved in Feasibility Analysis

The following steps are to be followed while performing feasibility analysis −

- Form a project team and appoint a project leader.
- Develop system flowcharts.
- Identify the deficiencies of current system and set goals.
- Enumerate the alternative solution or potential candidate system to meet goals.

- Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
- Weight the performance and cost effectiveness of each candidate system.
- Rank the other alternatives and select the best candidate system.
- Prepare a system proposal of final project directive to management for approval.



3.1TECHNICAL FEASIBILITY

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project when writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problem

At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost).

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

3.2OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, predictability, disposability, sustainability,

affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors' are to be realized.

A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design.

Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

3.3BEHAVIORAL FEASIBILITY

It evaluates and estimates the user attitude or behavior towards the development of new system.

It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. [t is common knowledge that computer installations have something to do with turnover, transfers, retraining, and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

3.4 Economic Feasibility

- It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
- It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
- The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
- It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

3.5TIME FEASIBILITY

A time feasibility study will take into account the period in which the project is going to take up to its completion. A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Time feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

3.5 FINANCIAL FESIBILITY

In case of a new project, financial viability can be judged on the following parameters:

- Total estimated cost of the project
- Financing of the project in terms of its capital structure, debt to equity ratio and promoter's share of total cost
- Existing investment by the promoter in any other business
- Projected cash flow and profitability The financial viability of a project should provide the following information:[12]
- Full details of the assets to be financed and how liquid those assets are.
- Rate of conversion to cash-liquidity (i.e., how easily the various assets can be converted to cash).
- Project's funding potential and repayment terms.
- Sensitivity in the repayments capability to the following factors:
- Mild slowing of sales.
- Acute reduction/slowing of sales.
- Small increase in cost.
- Large increase in cost.
- Adverse economic condition.

3.6 Schedule Feasibility

It is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then its schedule feasibility is appraised as high. In many cases a project will be unsuccessful if it takes longer than it was estimated: some external environmental conditions may change, hence a project can lose its benefits, expediency and profitability. If a work to be accomplished at a project does not fit the timeframes demanded by its customers, then a schedule is unfeasible (amount of work should be reduced or other schedule compression methods applied).

If the project managers want to see their projects completed before they can lose their utility, they (project managers) need to give proper attention to controlling their schedule feasibility: to calculate and continually reexamine whether it is possible to complete all amount and scope of work lying ahead, utilizing the given amount of resources, within required period of time. Schedule feasibility study includes use of the following matters:

- Project Estimation;
- Gantt and PERT charts;
- CPM (Critical Path Method);
- Change Management;

CHAPTER 4

SYSTEM SPECIFICATION.

A **System Requirements Specification** (abbreviated SysRS when need to be distinct from a software requirements specification (SRS)) is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers.

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) **system requirements** and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of System requirements, is a generalization of this first definition, giving the requirements to be met in the design of a system or sub-system. Typically an organization starts with a set of Business requirements and then derives the System requirements from there.

4 (i) Hardware requirements:

(a)RAM (1GB or above)

(b)Pentium IV or above processor

(c)Hard disk (20GB with free space)

(d)VRAM_PCI, VGA(32MB)

4(a).RAM

Random-access memory (**RAM**) is a form of computer data storage that stores data and machine code currently being used. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, CD-RWs, DVD-RWsand the older magnetic tapes and drum memory, the time required to read and write data items varies significantly depending on their physical locations on the recording medium, due to mechanical limitations such as media rotation speeds and arm movement.

RAM contains multiplexing and demultiplexing circuitry, to connect the data lines to the addressed storage for reading or writing the entry. Usually more than one bit of storage is accessed by the same address, and RAM devices often have multiple data lines and are said to be "8-bit" or "16-bit", etc. devices.

In today's technology, random-access memory takes the form of integrated circuits. RAM is normally associated with volatile types of memory (such as DRAM modules), where stored information is lost if power is removed, although non-volatile RAM has also been developed. Other types of non-volatile memories exist that allow random access for read operations, but either do not allow write operations or have other kinds of limitations on them. These include most types of ROM and a type of flash memory called *NOR-Flash*.

Integrated-circuit RAM chips came into the market in the early 1970s, with the first commercially available DRAM chip, the Intel 1103, introduced in October 1970.

4(b).PENTIUM-IV PROCESSOR:

Pentium-iv is a brand by Intel for an entire series of single-core CPUs for desktops, laptops and entry-level servers. The processors were shipped from November 20, 2000, until August 8, 2008.

All Pentium 4 CPUs are based on the Net Burst architecture. The Pentium 4 *Willamette* (180 nm) introduced SSE2, while the *Prescott* (90 nm) introduced SSE3. Later versions introduced Hyper-Threading Technology (HTT).

The first Pentium 4-branded processor to implement 64-bit was the *Prescott* (90 nm) (February 2004), but this feature was not enabled. Intel subsequently began selling 64-bit Pentium 4s using the *"E0" revision* of the Prescott's, being sold on the OEM market as the Pentium 4, model F. The E0 revision also adds execute Disable (XD) (Intel's name for the NX bit) to Intel 64. Intel's official launch of Intel 64 (under the name EM64T at that time) in mainstream desktop processors was the N0 stepping Prescott-2M.

4(c).Hard disk:

A **hard disk drive** (**HDD**), **hard disk**, **hard drive**, or **fixed disk**,is an electromechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material. The platters are paired with magnetic heads, usually arranged on a moving actuator arm, which read and write data to the platter surfaces. Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order and not only sequentially. HDDs are a type of non-volatile storage, retaining stored data even when powered off.

Introduced by IBM in 1956, HDDs became the dominant secondary storage device for general-purpose computers by the early 1960s. Continuously improved, HDDs have maintained this position into the modern era of servers and personal computers. More than 200 companies have produced HDDs historically, though after extensive industry consolidation most units are manufactured by segues, Toshiba, and Western Digital. HDDs dominate the volume of storage produced (extra bytes per year) for servers. Though production is growing slowly, sales revenues and unit shipments are declining because solid-state drives (SSDs) have higher data-transfer rates, higher areal storage density, better reliability, and much lower latency and access times.

The revenues for SSDs, most of which use NAND, slightly exceed those for HDDs. Though SSDs have nearly 10 times higher cost per bit, they are replacing HDDs in applications where speed, power consumption, small size, and durability are important.

The primary characteristics of an HDD are its capacity and performance. Capacity is specified in unit prefixes corresponding to powers of 1000: a 1-terabyte (TB) drive has a

capacity of 1,000 gigabytes (GB; where 1 gigabyte = 1 billion bytes). Typically, some of an HDD's capacity is unavailable to the user because it is used by the file system and the computer operating system, and possibly inbuilt redundancy for error correction and recovery. Also there is confusion regarding storage capacity, since capacities are stated in decimal Gigabytes (powers of 10) by HDD manufacturers, whereas some opera ring systems report capacities in binary Gibbets, which results in a smaller number than advertised. Performance is specified by the time required to move the heads to a track or cylinder (average access time) adding the time it takes for the desired sector to move under the head (average latency, which is a function of the physical rotational speed in revolutions per minute), and finally the speed at which the data is transmitted (data rate).

4(d).VRAM:

Video RAM, or **VRAM**, is a dual-ported variant of dynamic RAM (DRAM), which was once commonly used to store the frame buffer in graphics adapters.

It was invented by F. Dill, D. Ling and R. Matick at IBM Research in 1980, with a patent issued in 1985 (US Patent 4,541,075). The first commercial use of VRAM was in a high-resolution graphics adapter introduced in 1986 by IBM for its RT PC system, which set a new standard for graphics displays. Prior to the development of VRAM, dual-ported memory was quite expensive, limiting higher resolution bitmapped graphics to high-end workstations. VRAM improved the overall frame buffer throughput, allowing low cost, high-resolution, high-speed, color graphics. Modern GUI-based operating systems benefitted from this and thus it provided a key ingredient for proliferation of graphical user interfaces (GUIs) throughout the world at that time.

VRAM has two sets of data output pins, and thus two ports that can be used simultaneously. The first port, the DRAM port, is accessed by the host computer in a manner very similar to traditional DRAM. The second port, the video port, is typically read-only and is dedicated to providing a high throughput, serialized data channel for the graphics chipset.

Typical DRAM arrays normally access a full row of bits (i.e. a word line) at up to 1,024 bits at one time, but only use one or a few of these for actual data, the remainder being discarded. Since DRAM cells are destructively read, each row accessed must be sensed, and re-written. Thus, 1,024 sense amplifiers are typically used. VRAM operates by not discarding the excess bits which must be accessed, but making full use of them in a simple way. If each horizontal scan line of a display is mapped to a full word, then upon reading one word and latching all 1,024 bits into a separate row buffer, these bits can subsequently be serially streamed to the display circuitry. This will leave access to the DRAM array free to be accessed (read or write) for many cycles, until the row buffer is almost depleted. A complete DRAM read cycle is only required to fill the row buffer, leaving most DRAM cycles available for normal accesses.

4(ii) Software requirement:

(a) : window Operating system xp,7,10

(b) :Application server: apache tomcat

4(ii)(a) Operating system:

Windows 7 is a personal computer operating system that was produced by Microsoft as part of the Windows NT family of operating systems. It was released to manufacturing on July 22, 2009 and became generally available on October 22, 2009 less than three years after the release of its predecessor, Windows Vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time.

Windows 7 was primarily intended to be an incremental upgrade to Microsoft Windows, intended to address Windows Vista's poor critical reception while maintaining hardware and software compatibility. Windows 7 continued improvements on Windows Aero (the user interface introduced in Windows Vista) with the addition of a redesigned taskbar that allows applications to be "pinned" to it, and new window management features. Other new features were added to the operating system, including libraries, the new file sharing system Home Group, and support for multi touch input. A new "Action Center" interface was also added to provide an overview of system security and maintenance information, and tweaks were made to the User Account Control system to make it less intrusive. Windows 7 also shipped with updated versions of several stock applications, including Internet Explorer 8, Windows Media Player, and Windows Media Center.

In contrast to Windows Vista, Windows 7 was generally praised by critics, who considered the operating system to be a major improvement over its predecessor due to its increased performance, its more intuitive interface (with particular praise devoted to the new taskbar), fewer User Account Control popup, and other improvements made across the platform. Windows 7 was a major success for Microsoft; even prior to its official release, pre-order sales for 7 on the online retailer Amazon.com had surpassed previous records. In just six months, over 100 million copies had been sold worldwide, increasing to over 630 million licenses by July 2012. As of December 2018, 35.55% of computers running Windows are running Windows 7.

4(ii)(b)Application server:

An **application server** is a software framework that provides both facilities to create web applications and a server environment to run them.

Application Server Frameworks contain a comprehensive service layer model. An application server acts as a set of components accessible to the software developer through a standard API defined for the platform itself. For Web applications, these components are usually performed in the same running environment as their web server(s), and their main job is to support the construction of dynamic pages. However, many application servers target much more than just Web page generation: they implement services like clustering, fail-over, and load-balancing, so developers can focus on implementing the business logic.

Apache Tomcat, often referred to as **Tomcat Server**, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and Web Socket, and provides a "pure Java" HTTP web server environment in which Java code can run.

CHAPTER 5

Language and technology used

The language and technology which we are going to use in project hospital management system are given below .Here in this section we mention front end technology, back end technology as well as database and required language.

Front end technology:

(a)HTML

(b)JAVA SCRIPT

(c) JSP (JAVA SERVER PAGES)

(a)HTML: Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML Elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags,* written using angle brackets. Tags such as <img/> and <input/> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

(b)JAVA SCRIPT:

JAVA SCRIPT: **JavaScript** often abbreviated as **JS**, is a high-level, interpreted programming language that conforms to the java script specification. It is a programming language that is characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles . It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the

language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms *Vanilla JavaScript* and *Vanilla JS* refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design. JavaScript was influenced by programming languages such as self and Scheme.

(c)JSP (JAVA SERVER PAGES): Java Server Pages (**JSP**) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but it uses the java programming language. A **Java Server Pages compiler** is a program that parses JSPs, and transforms them into executable Java Servlets. A program of this type is usually embedded into the application server and run automatically the first time a JSP is accessed, but pages may also be precompiled for better performance, or compiled as a part of the build process to test for errors.

Some JSP containers support configuring how often the container checks JSP file timestamps to see whether the page has changed. Typically, this timestamp would be set to a short interval (perhaps seconds) during software development, and a longer interval (perhaps minutes, or even never) for a deployed Web application. JSP can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java byte code rather than machine code. Like any other Java program, they must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well.

The Web container creates JSP implicit objects like request, response, session, application, configuration, page, page Context, out and exception. JSP Engine creates these objects during translation phase.

CHAPTER 6

Modules

6(i) ENQUIRY MODULE: Under enquiry module the enquiry process will be done. This module manages activities related to patients' enquiry, appointment enquiry and tariff enquiry. Here appointment means the patients' appointment to doctors. Here are some highlighted points are given which will be performed under enquiry module: (a) Patient enquiry

(b)Appointment enquiry

(c)Tariff enquiry

6(ii) PATIENT MODULE*:* Under this module patient module are given means this module manages activities to registration of patients and payment details of patients .Following points are given below which is related to patients:

(a)Registration

(b)Login

(c)Payment details

6(iii) BLOOD BANK: Under this module blood bank related issues are managed means manages activities related to blood that is how much blood are available (group of blood) in the lab and also addable blood and spendable blood. Following points are given below which is related to blood bank module are:

(a)Check availability

(b)Add blood

(c)Used blood

6(iv) EMPLOYEE MODULE: Here in this module, it manages the activities related to the registration, login, department details, salary details, and leave details of employee means all the staff member of hospital such as doctors, nurse, inpatient, outpatient, guard boy, and o on. Following points are given below:

(a)Registration

(b)Login

(c)Department details

(d)Salary detail

CHAPTER 7

SDLC(waterfall model)

SDLC (Waterfall model):

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The sequential phases in Waterfall model are −

(a) Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

(b) System Design − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

(c) Implementation − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

(d) Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

(e) Deployment of system − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

(f) Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Advantages of waterfall model:

(a) This model is simple and easy to understand and use.

(b) It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.

(c) In this model phases are processed and completed one at a time. Phases do not overlap.
(d) Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

Disadvantages of waterfall model:

(a) Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
(b) No working software is produced until late during the life cycle.
(c) High amounts of risk and uncertainty.
(d) Not a good model for complex and object-oriented projects.
(e) Poor model for long and ongoing projects.
(f) Not suitable for the projects where requirements are at a moderate to high risk of changing

CHAPTER 8

DFD

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

The four components of data flow diagrams.

External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as "Submit payment."

Data store: files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as "Orders."

Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like "Billing details.

5.1 Zero level DFD

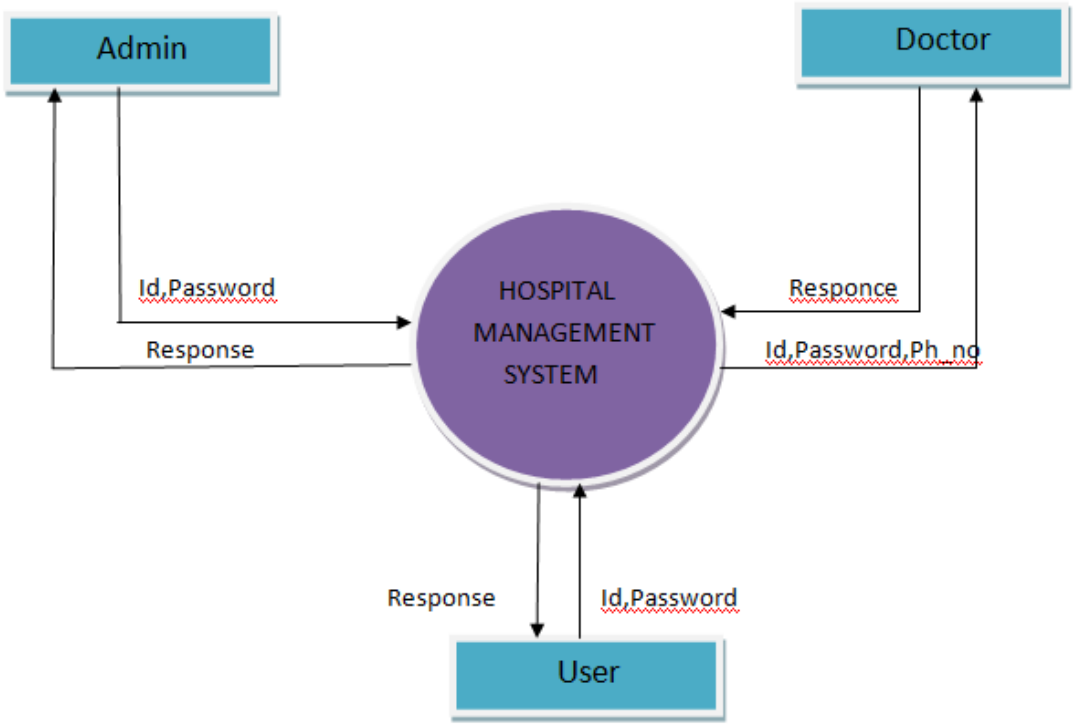


Fig.1

5.2 One Level Dfd

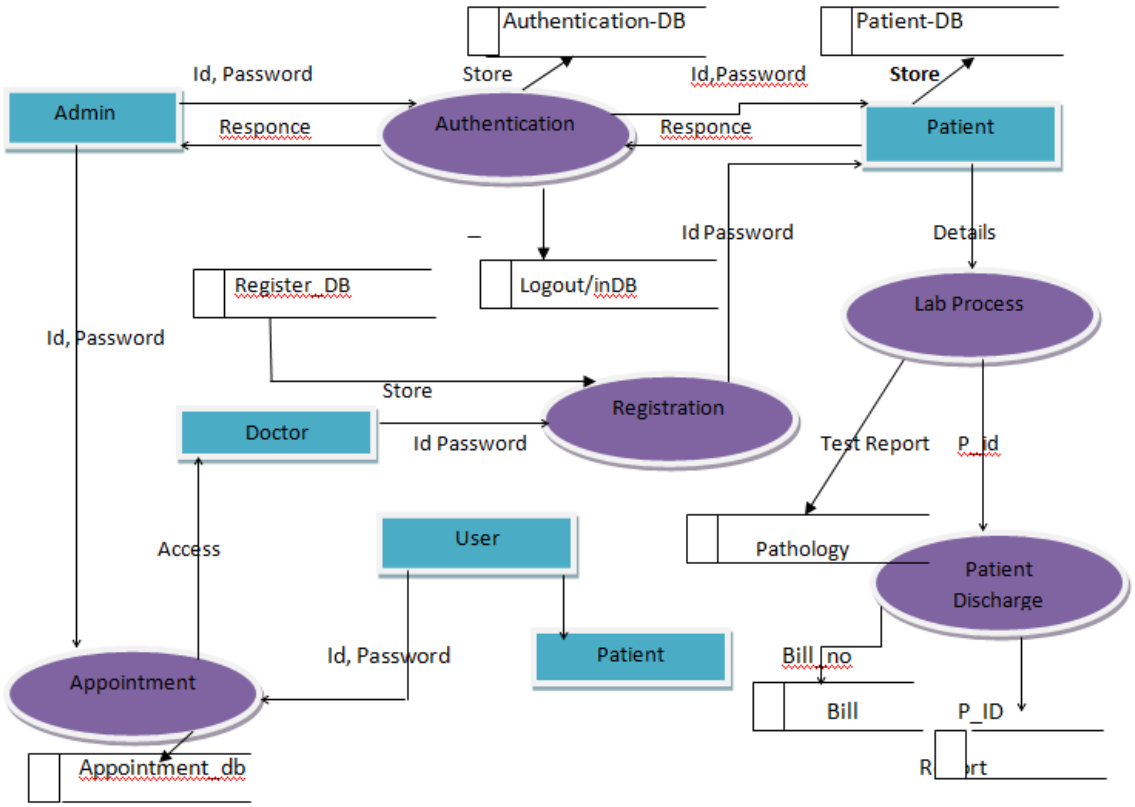


Fig.2

5.3 Two Level Dfd

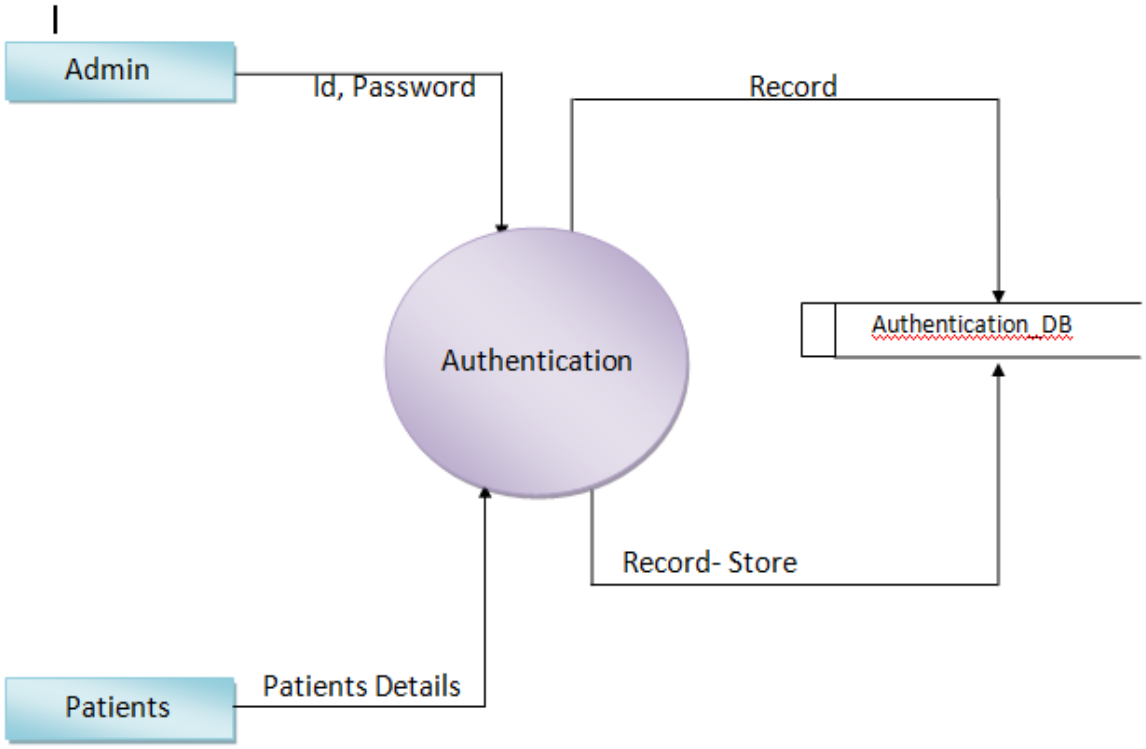


Fig.3

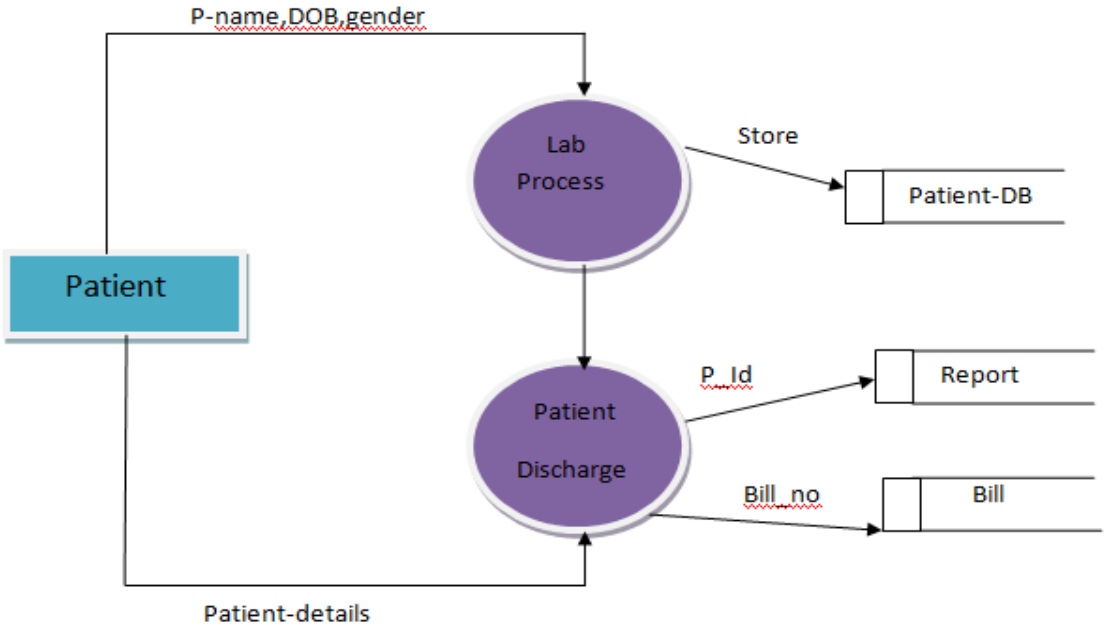

P-name,DOB,gender
Lab
Process
Store
Patient-DB
Patient
Patient
Discharge
P_Id
Report
Bill_no
Bill
Patient-details

Fig.4

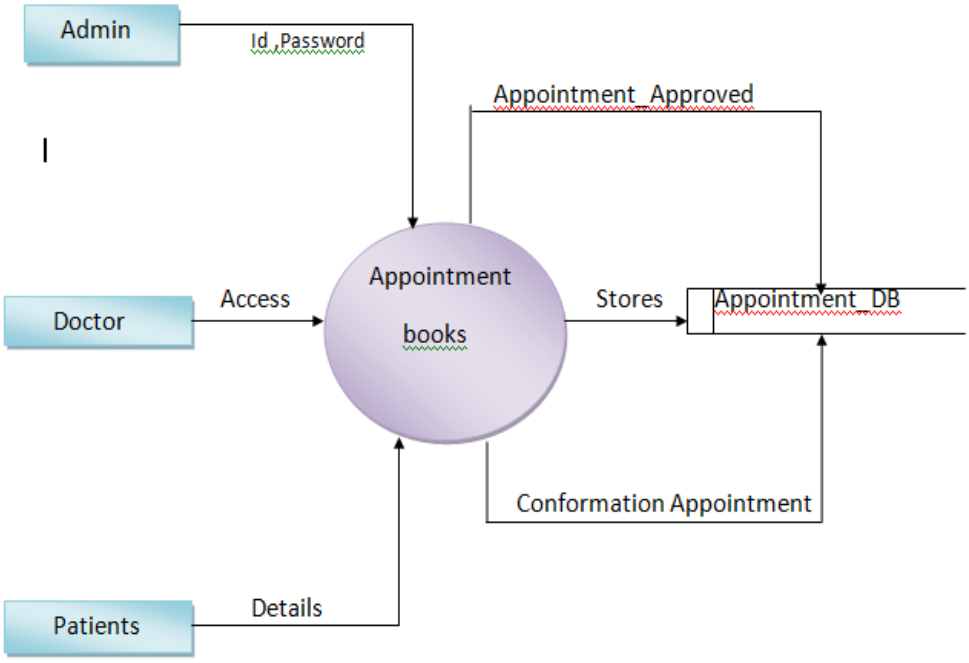

Admin
Id ,Password
Appointment_Approved
Doctor
Access
Appointment
books
Stores
Appointment_DB
Conformation Appointment
Patients
Details

Fig.5

CHAPTER 9

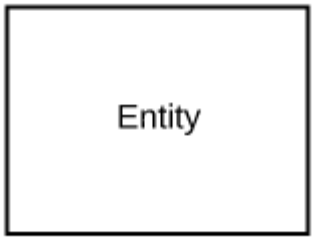
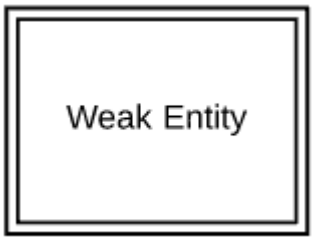
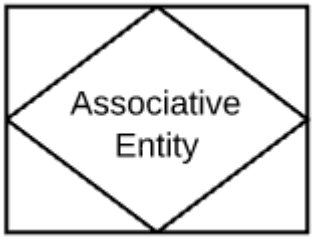
ER-DIAGRAM

ER-diagram:

Conceptual Data Models establish a broad view of what should be included in the model set. Conceptual ERDs can be used as the foundation for logical data models. They may also be used to form commonality relationships between ER models as a basis for data model integration. All of the symbols shown below are found in the UML Entity Relationship and Entity Relationship shape library of Lucid chart.

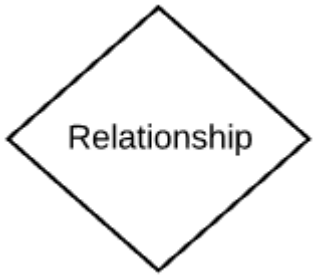
12(i) ERD entity symbols

Entities are objects or concepts that represent important data. Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams.

Entity Symbol	Name	Description
	Strong entity	These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity.
	Weak entity	Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity.
	Associative entity	Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances.

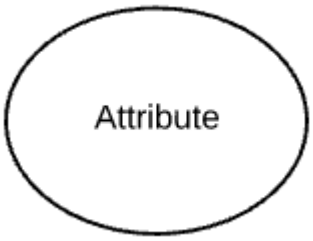
12(ii) ERD relationship symbols

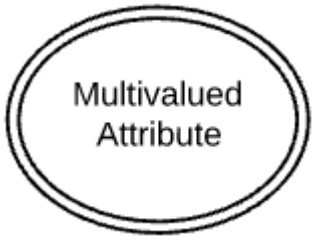
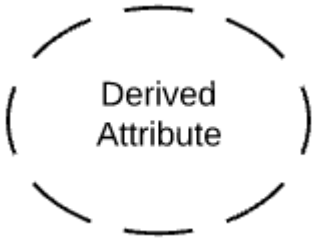
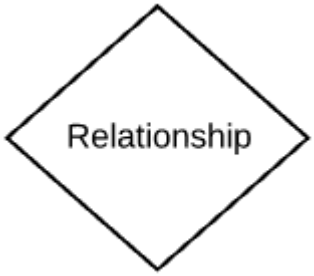
Within entity-relationship diagrams, relationships are used to document the interaction between two entities. Relationships are usually verbs such as assign, associate, or track and provide useful information that could not be discerned with just the entity types.

Relationship Symbol	Name	Description
	Relationship	Relationships are associations between or among entities.
	Weak relationship	Weak Relationships are connections between a weak entity and its owner.

12(iii) ERD Attribute symbols

ERD attributes are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.

Attribute Symbol	Name	Description
	Attribute	Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship.

Attribute Symbol	Name	Description
	Multi _valued attribute	Multi _valued attributes are those that are can take on more than one value.
	Derived attribute	Derived attributes are attributes whose value can be calculated from related attribute values.
	Relationship	Relationships are associations between or among entities.

ER DIAGRAM

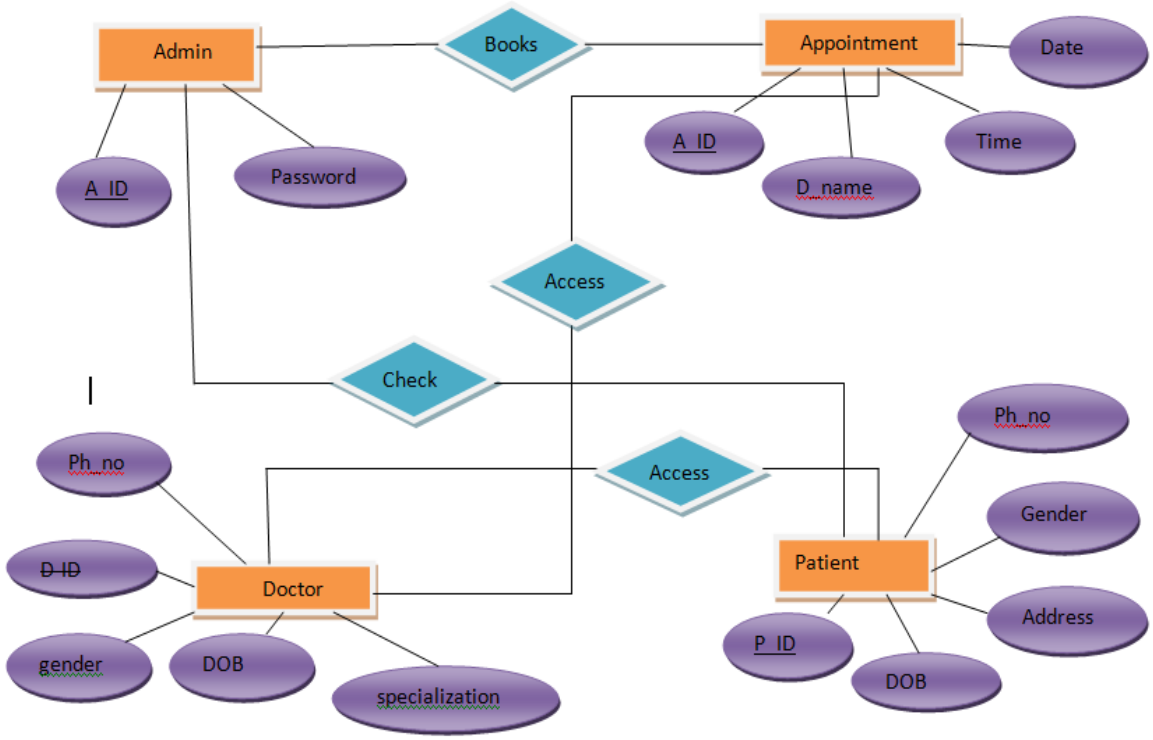


Fig.6

CHAPTER 10

Coding

1. Index Page HTML code:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
      <title>HOSPITAL MANAGEMENT SYSTEM
      </title>
      <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
<header>
      <nav>
<h1> AIM hospital</h1>
<ul><id="1">
      <li><a class="homered" href="#">HOME</a></li>
      <li><a class="homeblack"href="about/about.jsp">ABOUT</a></li>
      <li><a
class="homeblack"href="specialities/specialities.jsp">SPECIALITIES</
a></li>
      <li><a
class="homeblack"href="enquiry/enquiry.jsp">ENQUIRY</a></li>
      <li><a
class="homeblack"href="blood_bank/check_availability.jsp">BLOOD
BANK</a></li>
```

```
      <li><a class="homeblack"href="contact_us/contact.jsp">CONTACT
US</a></li>
      <li><a class="homeblack"href="login/login.jsp">LOGIN</a>



      </nav>


</header>
<div class="divider"></div>
<div class="fwimage1"> </div>



</body>


</html>
```

2. Index Page CSS code

```
@CHARSET "ISO-8859-1";
@import
url('https://fonts.googleapis.com/css?family=Oswald:400,700');



body{
      margin:0;
}
header{
```

```
        background: black;
        color: white;
        padding: 8px 8px 6px 40px;
        height: 50px;
    }
    header h1{
        display: inline;
        font-family: 'Oswald', sans-serif;
        font-weight: 400;
        font-size: 32px;
        float: left;
        margin-top: 0px;
        margin-right: 20px;

    }
    nav ul{
        display: inline;
        padding: 0px;
        float: left;
    }
    nav ul li{
        display: inline-block;
        list-style-type: none;
        color: white;
        float: left;
        margin-left: 80px;
```

```
}
nav ul li a{
	color: white;
	text-decoration: none;

}
#1{
	font-family: 'Chivo', sans-serif;
}
.homered {
	background-color: blue;
	padding: 23px 10px 22px 10px;
}
.divider{
	background-color: red;
	height: 10px;
}
.homeblack:hover {
	background-color: green;
	padding: 23px 10px 22px 10px;
}
.fwimage1{
	background-image: url(../img/image4.jpg);
	width: 100%;
	height: 100%;
	background-repeat: no-repeat;
	background-size: cover;
```

```
        margin: 0;
        padding: 0;

}

}
```

3.Add/Supply Blood HTML code

```
<!DOCTYPE html>
<html>
<head>
        <title>ADD/REMOVE OF BLOOD</title>
        <link rel="stylesheet" type="text/css"
href="../css/add_supplyblood.css">
</head>
<body>
        <center><h1>Add and Remove Blood</h1></center>
        <center><form action="/hms/add supply blood" method="post">
            <h3>Blood Group </h3><br>
            <select name="blood group">
<option value="A+">A+</option>
<option value="A-">A-</option>
<option value="B+">B+</option>
<option value="B-">B-</option>
<option value="AB+">AB+</option>
<option value="AB-">AB-</option>
<option value="O+">O+</option>
```

```
<option value="O-">O+</option>
            </select><br><br>
            <h3>Unit</h3><br>
            <input type="number" name="unit"><br><br>
            <input id="in" type="submit" name="blood form"
value="IN">
              <input id="out" type="submit"
name="blood form" value="OUT">
      </form>
</center>
</body>
</html>
```

4.Add/Supply Blood CSS code

```
@CHARSET "ISO-8859-1";
@import
url('https://fonts.googleapis.com/css?family=Oswald:400,700');

<style>
body{

      background:black
      background-size:cover;
      }
form{
      width:500px;
      margin: auto;
      padding: 20px;
      background:green;
```

```
        font-size: 20px;
        font-family: Verdana;
        color: yellow;
    }
    input[type=number]{
        width:80px;
        padding: 5px;
        font-size: 18px;
    }
    input[type=submit]
    {
    padding:5px;
    font-size: 20px;
    width:100px;
    color:blue;
    }
    select{
        font-size: 20px;
        padding: 5px;
        background:red;
    }
    #in:hover{
        height:3rem;
        width:5rem;
        color:black;
        background:red;
        box-shadow: 2rem, 2rem, 2rem black;
```

```
}
#out:hover{
	height:3rem;
	width:5rem;
	color:black;
	background:red;
	box-shadow: 2rem, 2rem, 2rem black;
}
</style>
5.Check Availability Code
<%@page import="java.sql.ResultSet"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ page import="com.hms.*"%>
<%@ page import="java.sql.*"%>
<html>
<head>
	<title>HOSPITAL MANAGEMENT SYSTEM
	</title>
	<link rel="stylesheet" type="text/css"
href="../css/admin.css">
</head>
<body>
<header>
	<nav>
<h1> AIM hospital</h1>
<%
```

```
      int
role=Integer.parseInt(session.getAttribute("user_role").toString());

      if(role==1){%>

<ul id="1">


    <li><a class="homered"
href="../login/admin.jsp">Admin_home</a></li>

      <li><a class="homeblack" href="#">Blood Bank</a></li>

      <li><a class="homeblack"href="hms.html">List of the
employees</a></li>

      <li><a class="homeblack"href="hms.html">List of the
patients</a></li>


      </ul><%}%>

<%

if(role==2){%>

<ul id="1">


    <li><a class="homered" href="../login/emp.jsp">Emphome</a></li>

      <li><a class="homeblack" href="#">Blood Bank</a></li>

      <li><a class="homeblack"href="hms.html">Add Patient</a></li>

      <li><a class="homeblack"href="hms.html">Employee
details</a></li>

      </ul>

<%} %>


      </nav>


</header>
```

```
<div class="divider"></div>
<div class="fwimage1"> </div><br><br><br><br>
<html>
<head>
	<title>AVAILABILITY OF BLOOD</title>
	<link rel="stylesheet" type="text/css" href="blood.css">
</head>
<body>
<table>
	<tr>
<th> S.n.</th>
<th> Group</th>
<th> Unit</th>

<th> Used/Supply</th>
</tr>
<% Oraclejdbc obj=new Oraclejdbc();
Statement stmt=obj.getStatement();
ResultSet rs=stmt.executeQuery("select * from bloodbank ");
 int sn=0;
while(rs.next()){%>
	<tr>
	<td> <%=sn+=1 %></td>
	<td> <%=rs.getString(1) %> </td>
	<td> <%=rs.getInt(2) %> </td>
		<td><%=rs.getInt(3) %></td>
</tr>
```

```
<% } %>


</table>

<br>

<center>

<a href="../blood bank/add supplyblood.jsp">

 <input id="addbutton" name="inputtext" type="button" value
="add/supply blood"/>

 </a>

</center>

</body>

</html>

</body>

</html>
```

6.Blood CSS Code

```
body{

background :url(../img/4.jpg);

}

table{

	width:800px;

	margin:auto;

	text-align:center;

	table-layout:fixed;

}
```

```
table, tr, th, td{
      padding:20px;
      color:white;
      border:1px solid #080808;
      border-collapse:collapse;
      font-size:18px;
      font-family:Arial;
      background:linear-gradient(top, #3c3c3c 0%, #222222 100%);
      background:-webkit-linear-gradient(top, #3c3c3c 0%, #222222
100%);
}
td:hover{
      background:orange;
}
#addbutton{

      width:7.3rem;
 height:3rem;
 margin:auto;
 color:white;
 background: blue;
      text-align: center;
box-shadow: 2rem, 2rem, 2rem grey;
}
#addbutton:hover{
      height:3rem;
      width:10rem;
```

```
	color:black;
	background:red;
	box-shadow: 2rem, 2rem, 2rem black;

}
```

7.Add/Supply Blood Connectivity Code

```
package com.hms;

import java.io.IOException;
import java.sql.*;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class AddSupplyBlood
 */
@WebServlet("/AddSupplyBlood")
public class AddSupplyBlood extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
```

```
    public AddSupplyBlood() {
        super();
        // TODO Auto-generated constructor stub
    }

     /**
      * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
      */
     protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
           // TODO Auto-generated method stub
           response.getWriter().append("Served at:
").append(request.getContextPath());
     }

     /**
      * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
      */
     protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
           // TODO Auto-generated method stub
           String
bloodGroupName=request.getParameter("blood_group");
           int unit=Integer.parseInt(request.getParameter("unit"));
           String bloodForm=request.getParameter("blood_form");
           response.getWriter().append("submit value "+bloodForm);
           Oraclejdbc obj=new Oraclejdbc();
```

```
            Statement stmt=obj.getStatement();
            int databaseUnit=0;
            int total=0;
            try {
                  ResultSet rs=stmt.executeQuery("select unit from
bloodbank where group_name='"+bloodGroupName+"'");
                  while(rs.next()){
                        databaseUnit=rs.getInt(1);


                  }
                  if(bloodForm=="IN")
                        total=databaseUnit+unit;
                  else
                        total=databaseUnit-unit;
                  int i=stmt.executeUpdate("update bloodbank set
unit="+total+"where group_name='"+bloodGroupName+"'");
                        response.sendRedirect("/hms/login/admin.jsp");
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }


}
```

8.Login code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <html>
<head>
     <title>HOSPITAL MANAGEMENT SYSTEM
     </title>
     <link rel="stylesheet" type="text/css"
href="../css/login.css">


   <link rel="stylesheet" type "text/css" href="login.css">
  </head>
 <body>
 <div class="login-box">


       <h1> Login Here </h1>
 <form action="/hms/loginauth" method="post">
  <p> user_id</p>
 <input type="text" name="user_id" placeholder="Enter user id">
 <p> Password</p>
  <input type="password" name="password" placeholder="Enter
password">
<input type="submit" name="submit" value ="Login">
<a href= "#">Forget Password</a>
</form>
</div>
```

```
</body>
</html>
```

9.Login CSS Code

```
@CHARSET "ISO-8859-1";
@import
url('https://fonts.googleapis.com/css?family=Oswald:400,700');

body{
margin:0;
padding:0;
background:url(../img/4.jpg);
background-size:cover;

font-family:sans-serif;
}

.login-box{
width: 320px;
height:400px;
background:rgba(0,0,0,0.5);
color:#fff;
top:50%;
left:50%;
position:absolute;
transfrom:translate(-50%,-50%);
box-sizing:border-box;
padding:70px 30px;
```

```
}



h1{
 margin: 0;
padding:0 0 20px;
text_align:center;
font-size:22px;
}
.login-box p{
margin:0;
padding:0;
font_weight:bold;
}
.login-box input{
 width:100%;
margin-bottom:20px;
}
.login-box input[type="text"],input[type="password"]
{
border:none;
border-bottom:1px solid #fff;
background:transparent;
outline:none;
height:40px;
color:#fff;
```

```
font-size:16px;

}

   .login-box input[type="submit"]
{
 border:none;
outline:none;
height:40px;
background:#1c8adb;
color:#fff;
font-size:18px;
border-radius:20px;
}

.login-box input[type="submit"]:hover
{
 cursor:pointer;
background:#39dc79;
color:#000;
}

.login-box a{
text-decoration:none;
font-size:14px;
color:#fff;
}
```

```
.login-box a:hover
{
color:#39dc79;
}
}
```

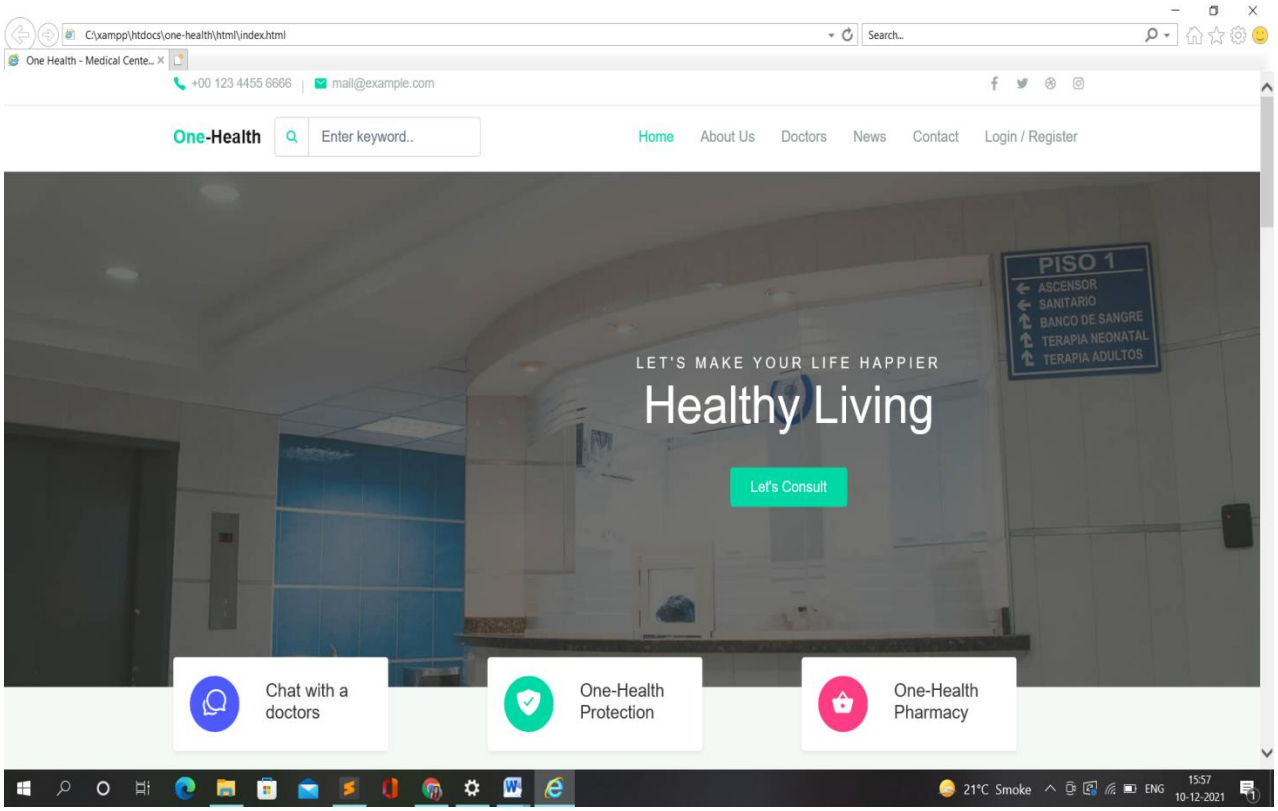
10.Login Connectivity Code

```
package com.hms;

import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class LoginAuth
 */
@WebServlet("/LoginAuth")
public class LoginAuth extends HttpServlet {
      private static final long serialVersionUID = 1L;
```

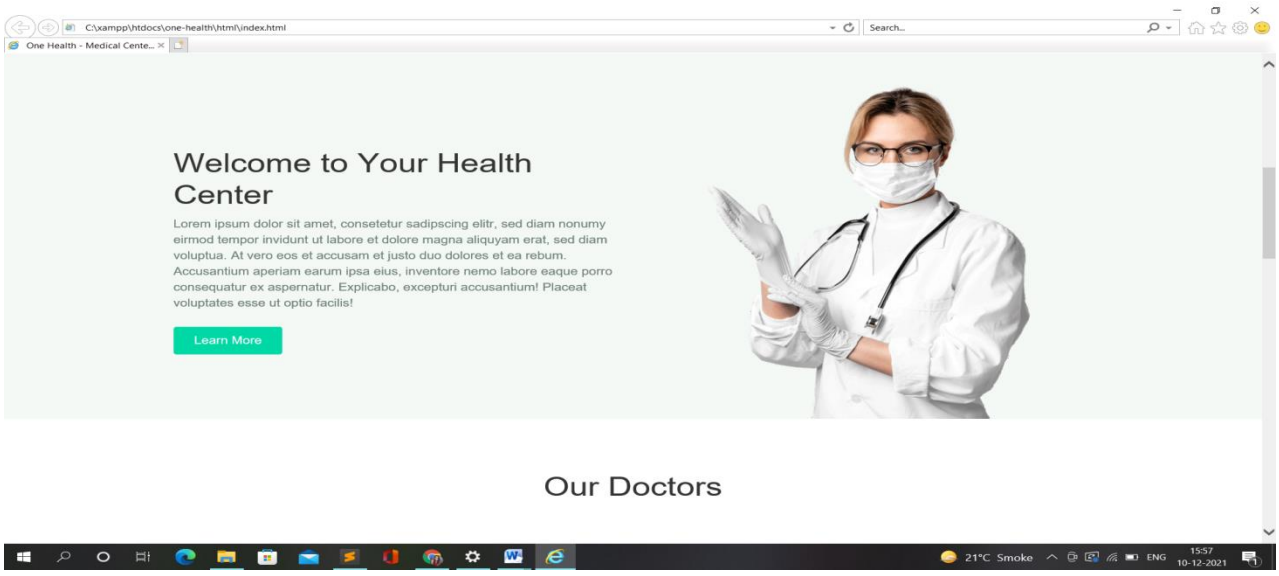
```
    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginAuth() {
        super();
        // TODO Auto-generated constructor stub
    }

     /**
      * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
      */
     protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
           // TODO Auto-generated method stub
           response.getWriter().append("Served at:
").append(request.getContextPath());
     }

     /**
      * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
      */
     protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
           // TODO Auto-generated method stub
           //doGet(request, response);
           String user_name=request.getParameter("user_id");
           String password=request.getParameter("password");
```

```
            Oraclejdbc obj=new Oraclejdbc();
            Statement stmt=obj.getStatement();
            try {
                  ResultSet rs=stmt.executeQuery("select user_role
from users where email = '" + user_name + "' and password = '"+
password+"'");
                  while(rs.next()){

      //response.getWriter().append(String.valueOf(rs.getInt(1)));
                        int user_role = rs.getInt(1);
                        HttpSession s=request.getSession();
                              s.setAttribute("user_role",user_role );

      response.getWriter().append(String.valueOf(rs.getInt(1)));
                        if(user_role == 1){

      response.sendRedirect("/hms/login/admin.jsp");
                        }
                        if(user_role==2){

      response.sendRedirect("/hms/login/emp.jsp");
                        }
                        if(user_role==3){

      response.sendRedirect("/hms/login/patient.jsp");


                        }

                  }
```

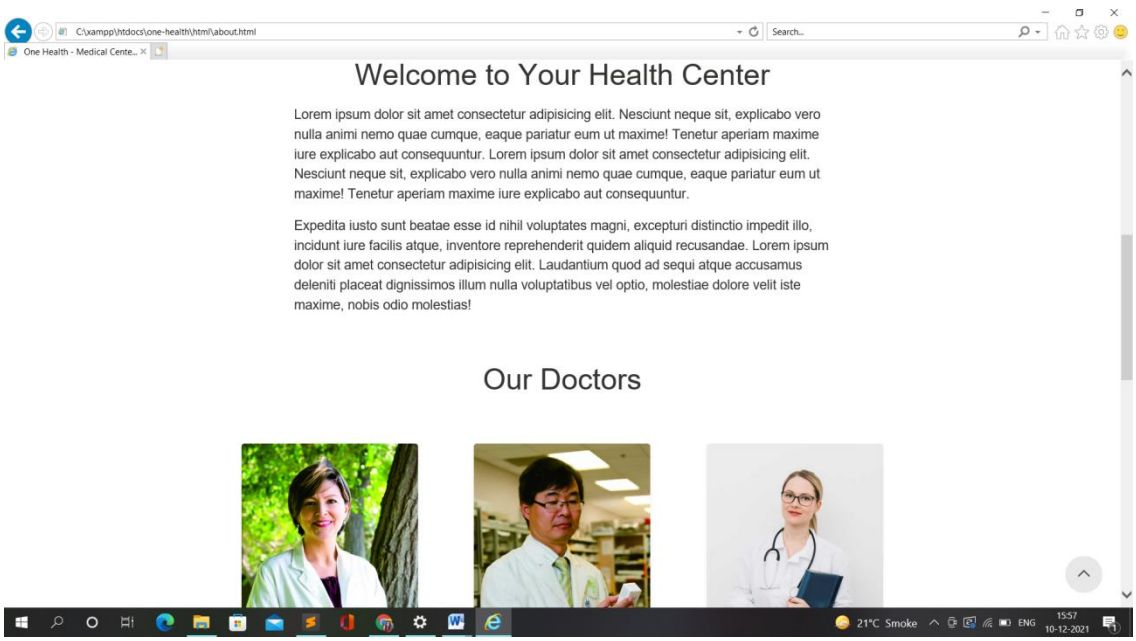
```
        }


                catch (SQLException e) {

                // TODO Auto-generated catch block
                e.printStackTrace();
                }
        }
}
```



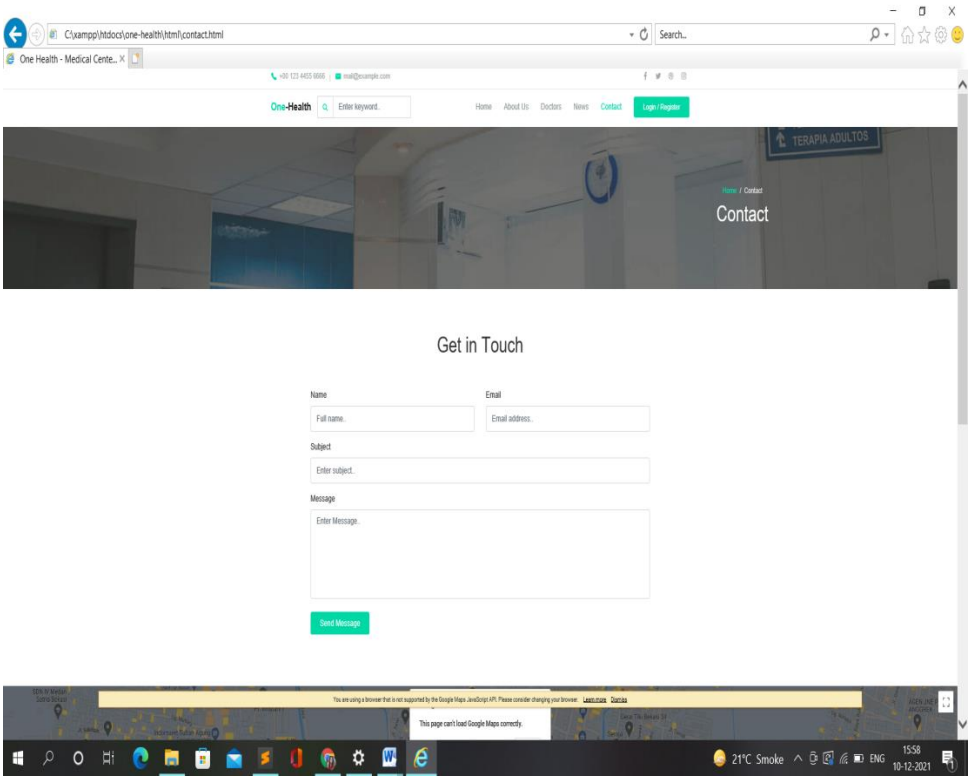
Screenshot.1



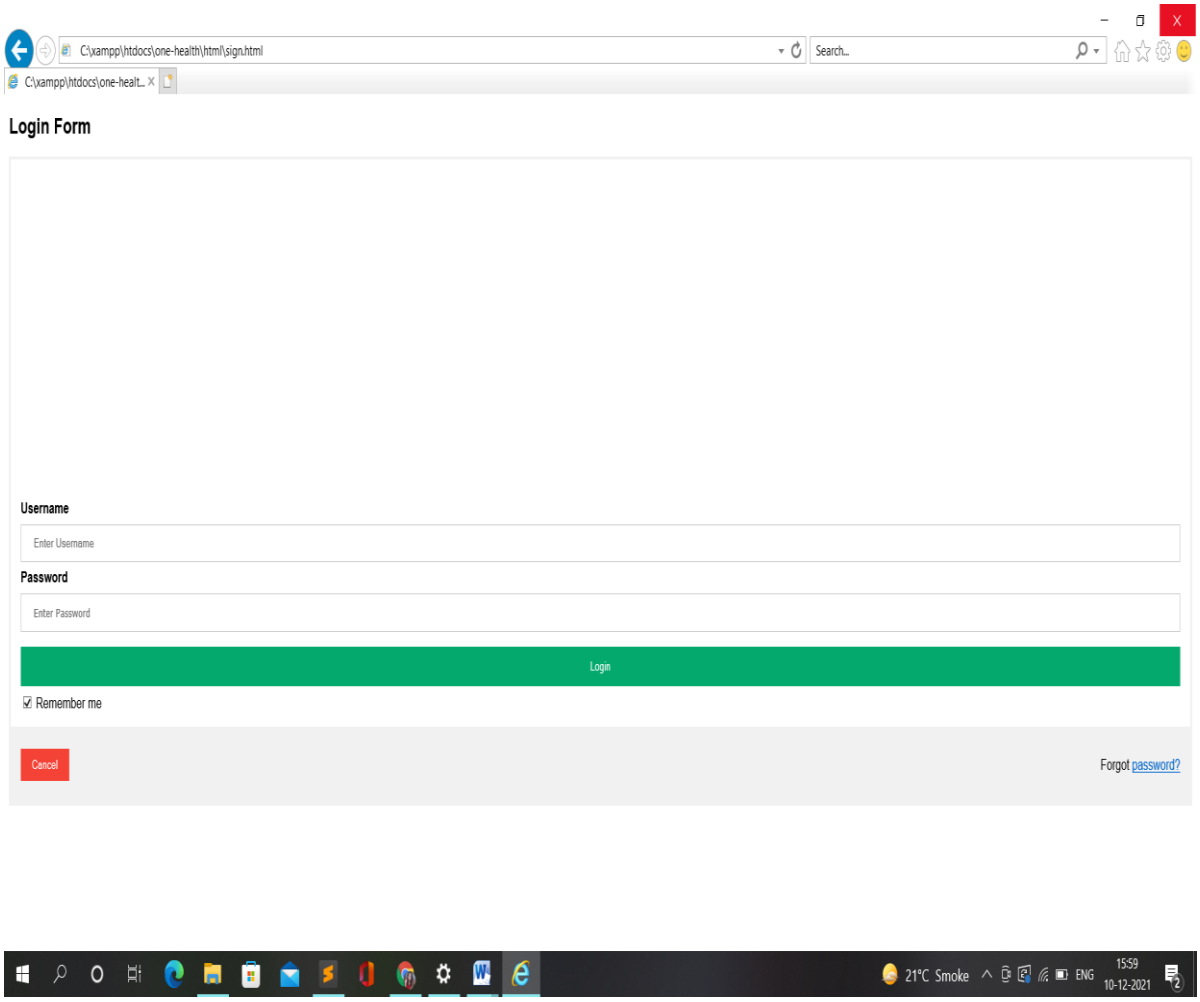
Screenshot.2



Screenshot.3



Screenshot.4



Screenshot.5

CHAPTER 11

Testing

Testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.

11(i) Static vs. dynamic testing:

There are many approaches available in software testing. Reviews, walkthroughs, or inspections are referred to as static testing, whereas executing programmed code with a given set of test cases is referred to as dynamic testing.

Static testing is often implicit, like proofreading, plus when programming tools/text editors check source code structure or compilers (pre-compilers) check syntax and data flow as static program analysis. Dynamic testing takes place when the program itself is run. Dynamic testing may begin before the program is 100% complete in order to test particular sections of code and are applied to discrete functions or modules. Typical techniques for these are either using stubs/drivers or execution from a debugger environment.

11(ii) White-box testing:

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing) verifies the internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system (the source code), as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g., in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

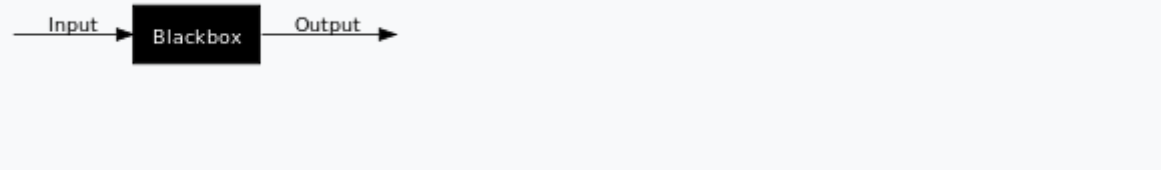
- API testing – testing of the application using public and private APIs (application programming interfaces)
- Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
- Fault injection methods – intentionally introducing faults to gauge the efficacy of testing strategies
- Mutation testing methods
- Static testing methods

Code coverage tools can evaluate the completeness of a test suite that was created with any method, including black-box testing. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Code coverage as a software metric can be reported as a percentage for:

- Function coverage, which reports on functions executed
- Statement coverage, which reports on the number of lines executed to complete the test
- Decision coverage, which reports on whether both the True and the False branch of a given tens.
- it has been executed

100% statement coverage ensures that all code paths or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly. Pseudo-tested functions and methods are those that are covered but not specified (it is possible to remove their body without breaking any test case).

11(iii) Black-box testing



Black-box testing (also known as functional testing) treats the software as a "black box," examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing, and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including

specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case or leaves some parts of the program untested.

This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

CHAPTER 12

Conclusion

This report will allow the users to store the details of doctors and patients of the hospital. This report package will allow storing the details of all the data related to the hospital. The system will be storing enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the hospital will considerably reduce data entry, time and also provide past information.

CHAPTER 13

Bibliography

(i)Introduction to c ++……by E. Balaguruswamy

(ii)Introduction to java……by MC Growhill

(iii) (i)Introduction to c ……..by E. Balaguruswamy

Website:

(i)www.tutorialponit.org

(ii)www.wikipedia.com

(iii)www.studytonight.com

(iv)www.javapoint.com