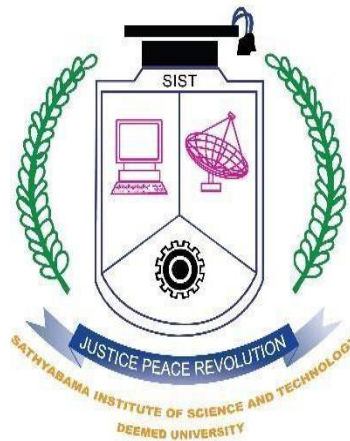


Prediction system for patient in Hospital management

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering
Degree in Computer Science and Engineering

By

DISHANEE GUHA (37110196)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL
OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

**JEPPIAARNAGAR, RAJIV GANDHI SALAI, C
HENNAI-600 119**

MARCH- 2021



INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAARNAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119R

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DISHANE E**

GUHA (37110196)

who carried out the project entitled "Prediction system for patient in Hospital management" under my supervision from November 2020 to April 2021.

Internal Guide

Mrs. Keerthi Samhitha

Head of the Department

Submitted for Vivavoce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, DISHANEE

GUHA(37110196) hereby declare that the project report entitled **Predictions system for patient in Hospital management** done by me under the guidance of **Mrs. Keerthi Samhitha** Department of CSE at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering in the department of Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAM A** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.S.Vigneshwari M.E., Ph.D. and Dr.L.Lakshmanan M.E., Ph.D. , Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. Keerthi Samhitha** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The purpose of the project entitled as “**Prediction system for patient in Hospitalmanagement**” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully. System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Assistant can be entered using a username and password. It is accessible either by an administrator or receptionist. The data can be retrieved easily. The data is well protected for personal use and makes the data processing very fast.

INDEX

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	ii
	LIST OFFIGURES	v
	LIST OF	v
	ABBREVIATIONSLIST	vi
	OFTABLES	
1	INTRODUCTION	1
	1.1OVERVIEW	1
	1.2EXSITINGSYSYTM	1
	1.3PROPOSEDSYSYTEM	2
	1.4OBJECTIVES	2
	1.5PROBLEMSTATEMENT	3
	1.6 SCOPE	3
	1.7 FEASIBILITY	4
	1.8SYSTEMREQUIREMENTS	7
	1.8.1HARDWAREREQUIREMENTS	
	1.8.2SOFTWAREREQUIREMENTS	
2	REQUIREMENTS	9
	2.1FUNCTIONALREQUIREMENTS	9
	SPECIFICATION	
	2.2NONFUNCTIONALREQUIREMENTS	9
	SPECIFICATION	
	2.2.1PERFORMANCE	10
	2.2.2SAFETYREQUIREMENTS	10
	2.2.3SECURITY	10
	2.2.4SOFTWAREQUALITYATTRIBUTES	10

3	TECHNOLOGYUSED	11
---	----------------	----

	3.1SOFTWARESPECIFICATION	11
4	SYSTEMIMPLEMENTATION	18
	4.1SYSTEMDESIGN	18
	4.1.1INTRODUCTIONTOUML	18
	4.2UMLAPPROACH	19
	4.2.1USECASEDIAGRAM	19
	4.2.2CLASSDIAGRAM	20
	4.2.3SEQUENCEDIAGRAM	22
	4.2.4COLLABORATIONDIAGRAM	23
	4.2.5DEPLOYMENTDIAGRAM	23
	4.2.6STATECHARTDIAGRAM	24
	4.2.7E-RDIAGRAM	25
	4.3DATABASETABLES	26
5	RESULTANDDISCUSSION	29
	5.1INTRODUCTION	29
	5.2SAMPLECODE	29
6	CONCLUSIONANDFUTUREENHANCEMENT	30
	REFERENCES	31
	APPENDICES	32
	A.SOURCECODE(EMBEDDED'C'&JAVA)	32
	B.SCREENSHOTS	42

List of Figures

DIAGRAM		PAGE NO
4.1	UseCaseDiagram	20
4.2	ClassDiagram	21
4.3	SequenceDiagram	22
4.4	CollaborationDiagram	23
4.5	DeploymentDiagram	24
4.6	State Chart Diagram	25
4.7	E-RDiagram	26

LISTOFABBREVIATIONS

ABBREVIATIONS

CSS

DBMS

FI

HCL

HTML

ICU

IP

JS

PHP

RDBMS

UML

XML

XUL

EXPANSION

CascadingStyleSheets

DatabaseManagementSystem

FormsInterpreter

Hardwarecompatibilitylist

HypertextMarkupLanguage

Intensivecareunit

InternetProtocoladdress

JavaScript

PersonalHomePage

RelationalDatabaseManagement

System

UnifiedModelingLanguage

ExtensibleMarkupLanguage

XMLUserInterfaceLanguage

ListofTables

TABLE		PAGENO
4.1	AdminTable	26
4.2	AppointmentTable	26
4.3	BillingTable	26
4.4	Doctor Table	27
4.5	PatientTable	27
4.6	PrescriptionTable	27
4.7	ServiceTypeTable	27
4.8	TreatmentTable	27
4.9	UserTable	28

1.INTRODUCTION

1.1 Overview:

The project Prediction system for patient in Hospital management 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. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and make the data processing very fast.

Hospital Assist 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-speciality hospitals, to cover a wider 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 Assist 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 Assist 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 Existing System:

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stored spread throughout the hospital management infrastructure. Often information is

incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that

no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

1.3 Proposed System:

The Hospital Assist is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

1.4 Objectives:

Admin Side

- To provide an easier and faster way to update details like test reports, prescriptions, bills, etc.
- To record details about all the available doctors, patients, and appointments
- To provide access to historic data of patients like previous appointments, medications, etc as well.

Doctor Side

- To provide an easier and faster way to access patient details like test reports, prescriptions, bills, etc, as well as historic data like previous appointments, medications, etc.
- To record details about all the patients and appointments
- To provide access to historic data of patients like previous appointments, medications, bills, etc as well.

Patient Side

- To enable easier and faster access to updated details like test reports, prescriptions, bills, etc.

- To provide access to historic data as well.

- To allow patients to book appointments easily.
- These are the various jobs that need to be done in a Hospital by the operational staff and doctors. All these works are done on papers. Digitalizing the process can reduce the use of paper and make access easy and secure.

1.5 Problem Statement:

Lack of immediate retrievals:-

The information is very difficult to retrieve and to find particular information like-
E.g. - To find out about the patient's history, the user has to go through various registers. This results in convenience and wastage of time.

Lack of immediate information storage:-

The information generated by various transactions takes time and efforts to be stored at a right place.

Lack of prompt updating:-

Various changes to information like patient details or immunization details of children are difficult to make as paper work is involved.

Error prone manual calculation: -

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient's bill based on various treatments.

Preparation of accurate and prompt reports:-

This becomes a difficult task as information is difficult to collect from various registers.

1.6 Scope:

- Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.

- Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they are summed up.
- Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
- Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
- Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and a lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

1.7 Feasibility:

Feasibility Study is a test of a system proposal according to its workability, impact on the organization, ability to meet needs and effective use of resources. It focuses on these major questions.

1. What are the user's demonstrable needs and how does a candidate system meet them?
2. What resources are available for given candidate system?
3. What are the likely impacts of the candidate system on the organization?
4. Whether it is worth to solve the problem?

During feasibility analysis for the project, following primary areas of interest are to be considered. Investigating and generating ideas about a new system does this.

Steps in Feasibility Analysis

1. Form a project team and appoint project leader.

2. Prepare system flowcharts.
3. Enumerate potential candidate systems.
4. Describe and identify characteristics of candidate system.
5. Determine and evaluate performance and cost effectiveness of each candidate system.
6. Weight system performance and cost data.
7. Select the best candidate system.
8. Prepare and report final project directive to management.

Technical Feasibility:

A study of resources available that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

- Can the work for the project be done with current equipment and existing software technologies?
- Can the system be upgraded if developed?
- If new technology is needed then what can be developed?

This is concerned with specifying equipment and software that will successfully satisfy the user requirements. The technical needs of the system may include:

Front-end and Back-end selection

An important issue for the development of a project is the selection of suitable front-end and back-end. When we decide to develop the project we want through an extensive study to determine the most suitable platform that suits the needs of the organization as well as help in development of the project.

The aspects of our study include the following factors.

Front-end Selection:

- It must have a graphical user interface that assist employees who are not from IT background.

- Scalabilityandextensibility
- Flexibility
- Robustness
- Accordingtotheorganizationrequirementandtheculture.
- Mustprovideexcellentreportingfeatureswithgoodprintingsupport.
- Platformindependent.
- Easyto debugandmaintain.

Back-endSelection:

- Multipleuser support.
- Efficientdatahandling.
- Provideinherentfeaturesforsecurity.
- Efficientdataretrievalandmaintenance.
- Storedprocedures.
- Popularity
- Operatingsystemcompatibility.
- Easytoinstall.

According to the above stated features we selected SQL server for the back-end. Thetechnical feasibility is frequently the most difficult area encountered at this stage. It isessential that the process of analysis and definition be conducted in parallel with anassessmenttotechnicalfeasibility.Itcentresontheexistingcomputersystem(hardware,s oftwareetc) andtowhatextentitcansupporttheproposedsystem.

EconomicFeasibility

Thisstudyiscarriedouttochecktheeconomicimpactwillhaveonthesystemwillhave on the organization. The amount of fund that the company can pour into theresearchanddevelopmentofthesystemislimited.Theexpendituresmustbejustified.Thus the developed system as well within the budget and this was achieved becausemostofthetechnologiesusedarefreelyavailable.Onlythecustomisedproductsha veto be purchased.

Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

1.8 Requirements:

1.8.1 Software Requirements:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software Requirements for Present Project:

- Operating System : Windows 7 and above
- Frontend : HTML, CSS, JavaScript.
- Server side script : PHP
- Database : MySQL

1.8.2 Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

Hardware Requirements for Present

Project:Processor :

IntelDualCorei3,orfasterprocessorRAM

: 4 GB

Hard Disk :20GBandNetworkConnectivity

2. REQUIREMENT AND SPECIFICATION

2.1 Functional Requirement Specification:

The following are the facilities provided by the system to the user.

- **Index/Home Page:**
The first screen that is displayed on entering the website. It contains a navigation bar which takes us to login page, online appointments, and contact details.
- **Login Using Credentials:**
There are different login pages for patient, doctor, and admin (receptionist/pharmacist / accountant). Each page requires entering the correct login id and password.
- **Online Appointment Booking:**
This page allows a patient to book an appointment through the website rather than physically going to the hospital.
- **View Appointments:**
This page allows an admin or a doctor to view all the pending and approved appointments of all the patients.

Functional Requirements are the functions that define the system. The system functions can be described as follows.

- Admin
- User
- Doctor

Admin (Receptionist/Pharmacist/Accountant) module:

- Manage department of hospitals, user, doctor accounts
- Watch appointment of doctors
- Watch transaction reports of patient payment
- Allot bed, ward, cabin for patients
- Keep record of patient operation
- Manage medicine categories

- Watch prescription of patient
- Provide medication to prescriptions
- Create invoice for payment
- Order invoice to patient
- Take cash payment
- Watch payment history of patients

User(Patient) module:

- View appointment list and status with doctors
- View prescription details
- View medication from doctor
- View doctor list
- View admit history like bed, ward, ICU, etc
- Manage own profile

Doctor module:

- Manage patient account opening and updating
- Create, manage appointment with patient
- Create prescription for patient
- Provide medication for patients
- Issue for operation of patients and create operation report
- Manage own profile

2.2 Non-functional Requirement Specification:

2.2.1 Performance

Response Time: The web pages shall give response within 8 seconds after registration of user.

Capacity: The web page must support as many users as needed.

2.2.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more recent state by reapplying or redoing the operations of committed transactions from the backup log, up to the time of failure.

2.2.3 Security

To login, the user must provide the following

- **Login ID:** A unique ID provided for each user.
- **Password:** A combination of various characters known to the user alone, which can be modified at any given time by the same user alone.

Modification

Any modification to the database should be synchronized.

2.2.4 Software Quality Attributes

- **Availability:** Since we are hosting our project on the server, it will be available all the time.
- **Correctness:**
The system should generate an appropriate report about different activities of the lab and should keep track of all records.
- **Maintainability:** The system should maintain correct schedules of labs and the documentation of all the lab equipment.
- **Usability:** The system should satisfy the maximum number of users' needs and give the response within seconds.

3. TECHNOLOGY USED

3.1 Software Specification

HTML: HTML or Hypertext Markup Language is the standard markup language used to create web pages.

HTML is written in the form of HTML elements consisting of tags enclosed in angular brackets (like <HTML>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example . The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

Cascading Style Sheets (CSS):

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone

specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colours, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content.

CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However, if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

MySQL:

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web; it runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on a number of platforms. The data in MySQL is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically.

Features of MySQL:

Internal and portability:

- Written in C and C++.

- Tested with a broad range of different compilers.
- Works on many different platforms.
- Tested with Purify (a commercial memory leak detector) as well as with Valgrind, a GPL tool.
- Uses multi-layered server design with independent modules.

Security:

- A privilege and password system that is very flexible and secure, and that enables host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

Scalability and Limits:

- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- Support for up to 64 indexes per table (32 before MySQL 4.1.2). Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 767 bytes for InnoDB tables, or 1000 for MyISAM; before MySQL 4.1.2, the limit is 500 bytes. An index may use a prefix of a column for CHAR, VARCHAR, BLOB, or TEXT column types.

Connectivity:

Clients can connect to MySQL Server using several protocols:

- Clients can connect using TCP/IP sockets on any platform.
- On Windows systems in the NT family (NT, 2000, XP, 2003, or Vista), clients can connect using named pipes if the server is started with the `--enable-named-pipe` option. In MySQL 4.1 and higher, Windows servers also support shared-memory connections if started with the `--shared-memory` option. Clients can connect through shared memory by using the `--protocol=memory` option.

Localization:

- **Clients and Tools:** The server can provide error messages to clients in many languages.
- All data is saved in the chosen character set.
- MySQL includes several client and utility programs. These include both command-line programs such as `mysqldump` and `mysqladmin`, and graphical programs such as `MySQL Workbench`.
- MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the `mysqlcheck` client. MySQL also includes `myisamchk`, a very fast command-line utility for performing these operations on MyISAM tables.
- MySQL programs can be invoked with the `--help` or `-?` option to obtain online assistance.

Why to use MySQL:

- Leading open source RDBMS
- Ease of use – No frills
- Fast
- Robust
- Security
- Multiple OS support
- Free
- Technical support
- Support large database – upto 50 million rows, file size limit upto 8 Million TB

JavaScript:

JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

Other uses of JavaScript:

- Delete HTML Elements
- Create new HTML Elements
- Copy HTML Elements
- In HTML, JavaScript is a sequence of statements that can be executed by the web browser.

JavaScript Statements:

- JavaScript statements are "commands" to the browser.
- The purpose of the statements is to tell the browser what to do.
- This JavaScript statement tells the browser to write "Hello Dolly" inside an HTML element with id="demo": Semicolon (;)
- Semicolon separates JavaScript statements.
- Normally you add a semicolon at the end of each executable statement.
- Using semicolons also makes it possible to write many statements on one line.

JavaScript Code:

- JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written.
- This example will manipulate two HTML elements:
- Example: `document.getElementById("demo").innerHTML="Hello Dolly";document.getElementById("myDIV").innerHTML="How are you?"`;

JavaScript Properties:

- Properties are the values associated with a JavaScript object.
- A JavaScript object is a collection of unordered properties.
- Properties can usually be changed, added, and deleted, but some are read-only.

PHP:

What is PHP?

- PHP is an acronym for "PHP Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language and PHP scripts are executed on the server.

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have an extension ".php"

What can PHP do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data

With PHP you are not limited to outputting HTML.

You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, MacOSX, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)

4. SYSTEM IMPLEMENTATION

4.1 System Design:

4.1.1 Introduction to UML:

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

- Visualizing
- Specifying
- Constructing
- Documenting

Visualizing

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building models that are precise, unambiguous and complete. UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Constructing

UML models can be directly connected to a variety of programming language

through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, design, source code, project plans, tests, prototypes, releases, etc...

4.2 UML Approach

UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. You draw a diagram to visualize a system from a different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes seven such diagrams:

- Use case diagram
- Class diagram
- Sequence diagram
- Collaboration diagram
- Deployment diagram
- State Chart diagram
- E-R diagram

Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of

actors, their goals (represented as use cases), and any dependencies between those use cases. Use case diagrams are formally included in two modeling languages defined by the OMG: the unified modeling language (UML) and the systems modeling language (SysML).

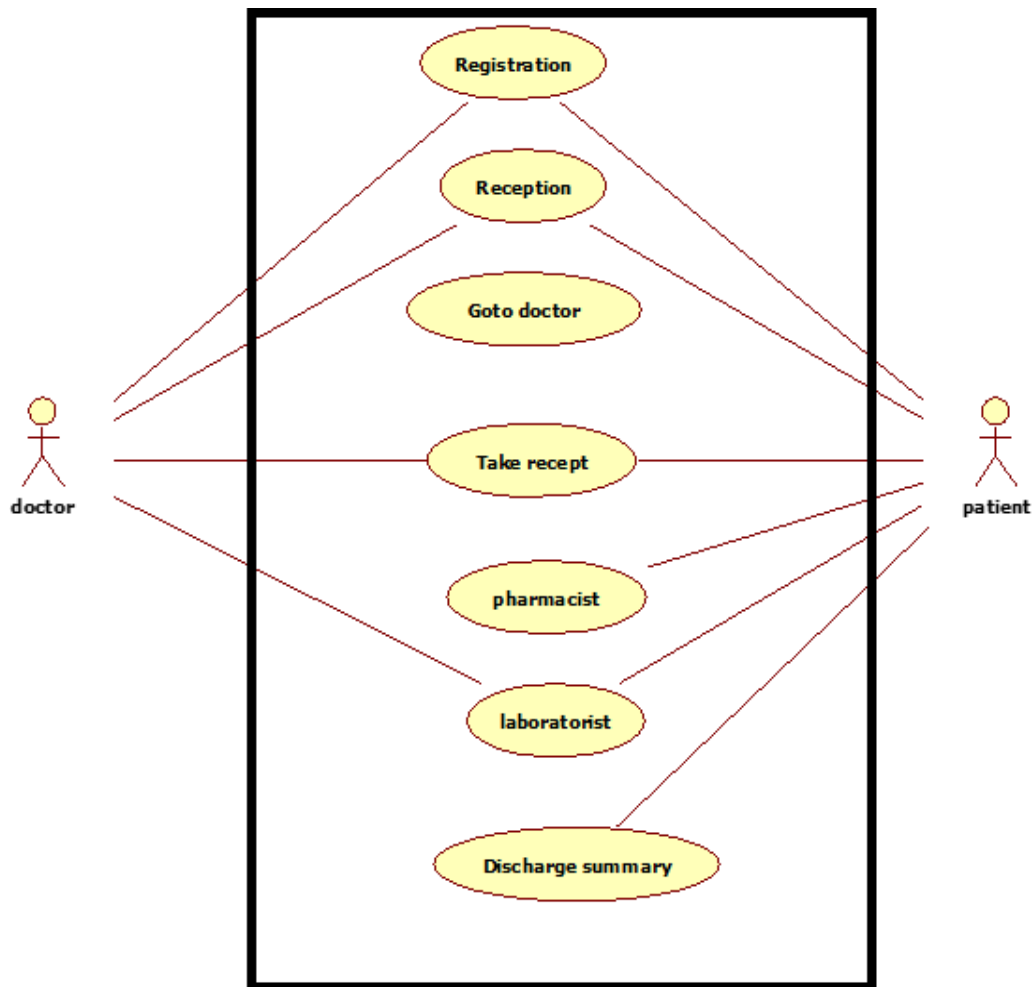


Fig4.1

4.2.2 ClassDiagram:

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class; it is divided into three areas. The uppermost area contains the name, the middle area contains the attributes and

the lowest are as show the operations. Class diagrams provide the representation that developers work from. Class diagrams help on the analysis side, too.

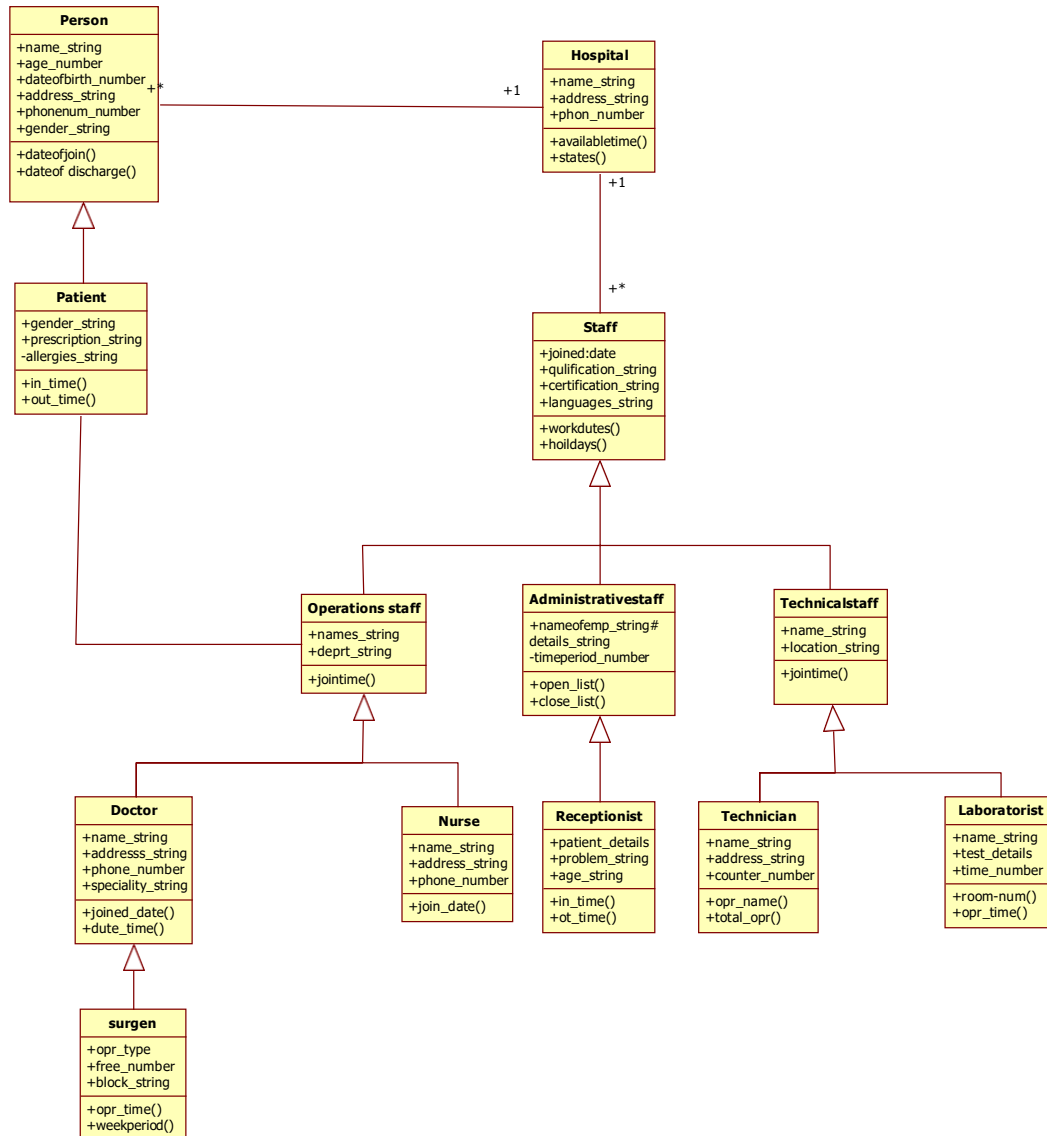


Fig4.2

4.2.3 SequenceDiagram:

A **SequenceDiagram** is an interaction diagram that emphasizes the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequencediagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

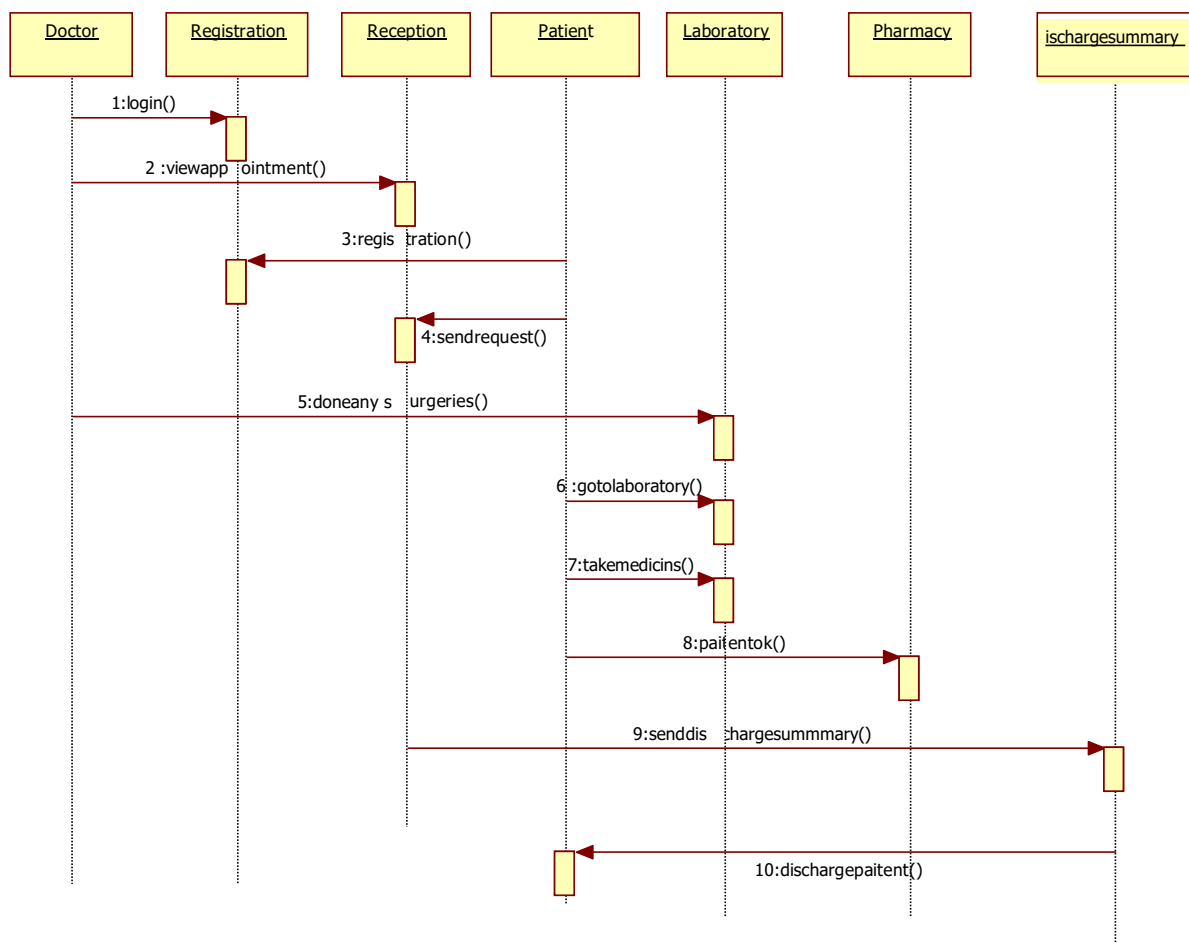


Fig4.3

4.2.4 CollaborationDiagram:

A **Collaboration Diagram** also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects. The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

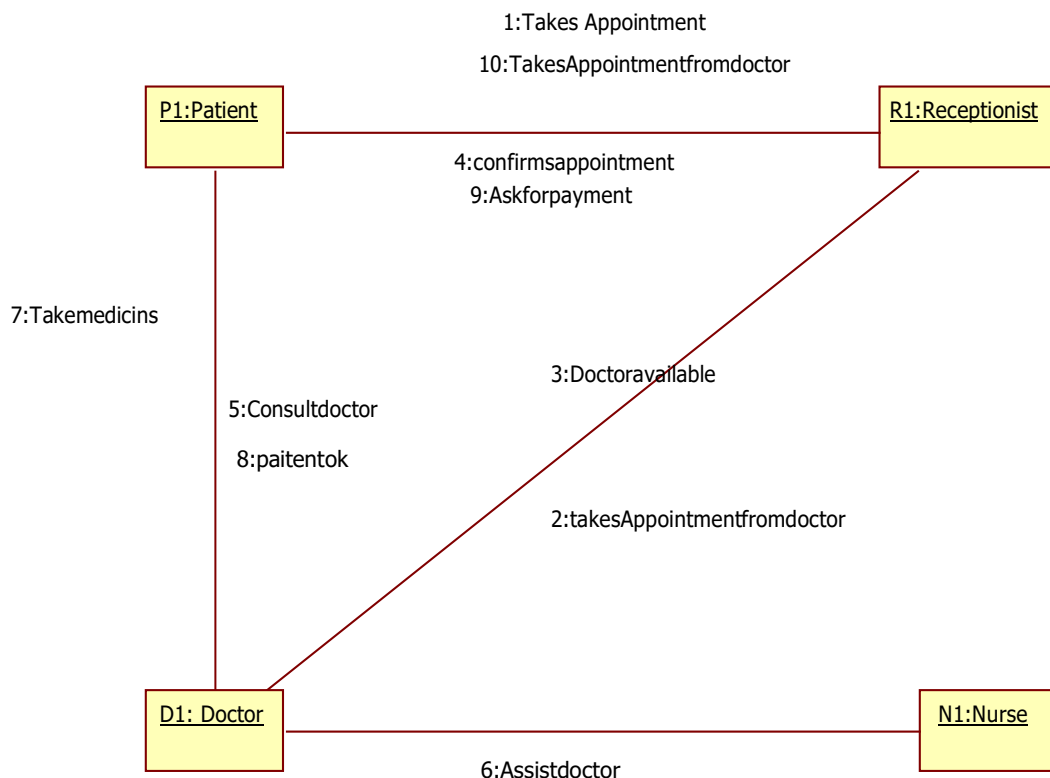


Fig4.4

4.2.5 DeploymentDiagram:

A **Deployment Diagram** shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components.

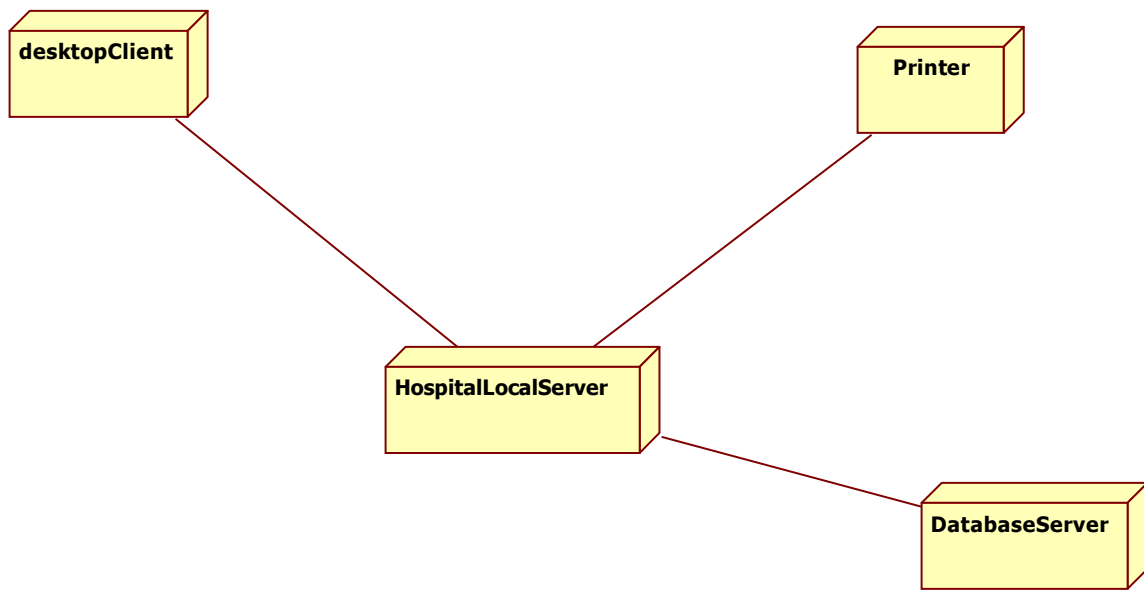


Fig4.5

4.2.6 StateChartDiagrams:

The state diagram shows the states of an object and represents activities as arrows connecting the states. The Activity Diagram highlights the activities. Each activity is represented by a rounded rectangle—narrower and more oval-shaped than the state icon. An arrow represents the transition from the one activity to the next. The activity diagram has a starting point represented by a filled-in circle, and an end point represented by a bullseye.

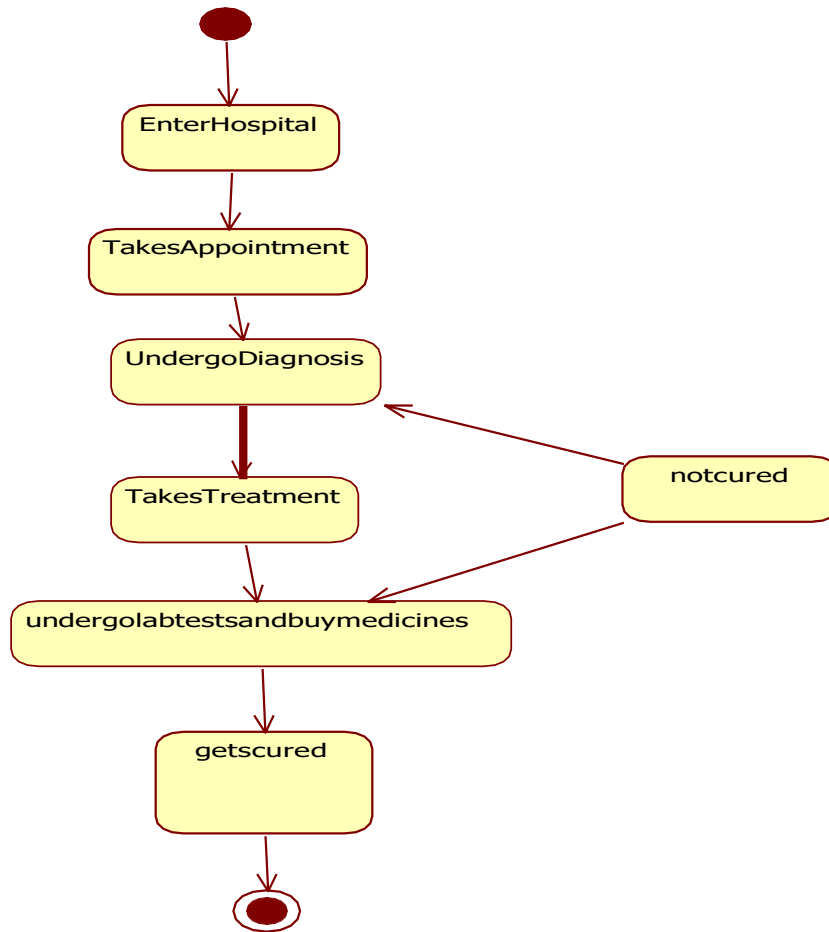


Fig4.6

4.2.7 E-RDiagram:

An **entity-relationship diagram (ERD)** is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An **ERD** is a conceptual and representational model of data used to represent the entity framework in a structure.

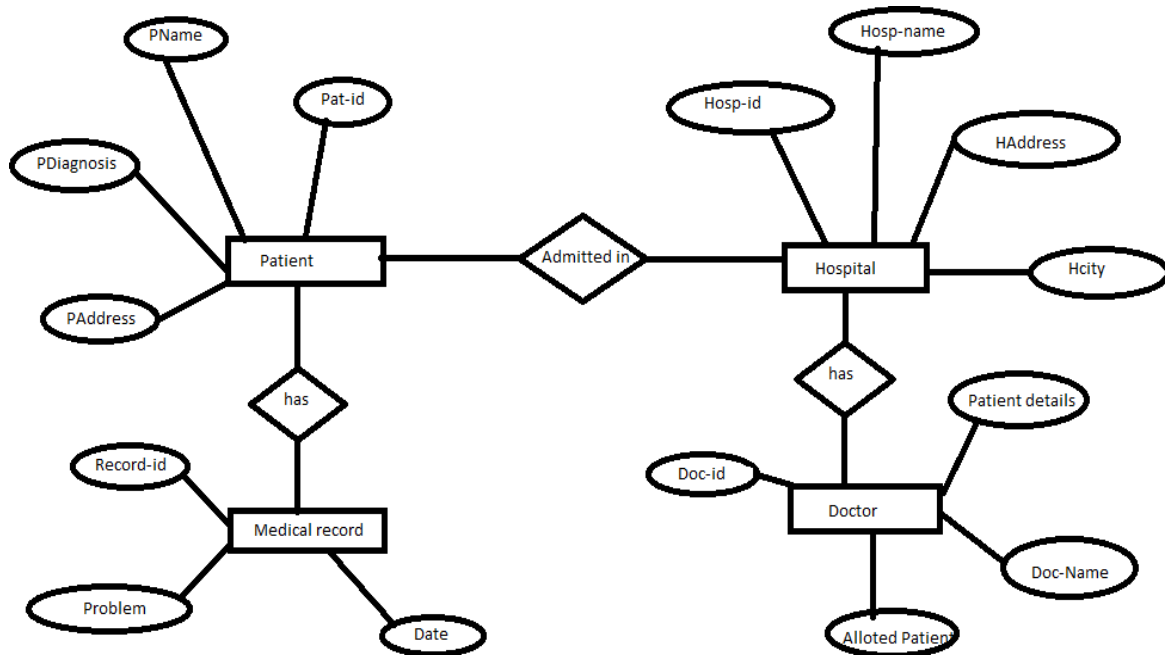


Fig.4.7

4.3 Database Tables:

4.1 AdminTable:

adminid	adminname	loginid	password	status	usertype
int(10)	varchar(25)	varchar(25)	varchar(25)	varchar(10)	varchar(30)

4.2 AppointmentTable:

appoint mentid	appoint menttype	patient id	room id	departmment id	appointment date	appointment time	doctorid	status	app_ reason
int(10)	varchar (25)	int(10)	int(10)	int(10)	date	time	int(10)	varchar (10)	text

4.3 BillingTable:

billingid	patient id	appoint mentid	billing date	billing time	discount	tax_amount	discount_r eason	discharge_t ime	discharge _date
int(10)	int(10)	int(10)	date	time	float(10,2)	float(10,2)	text	time	date

4.4 DoctorTable:

doctorid	doctor name	mobile no	depart mentid	loginid	passwor d	status	education	experience	consultan cy _charge
int(10)	varchar (50)	varchar (15)	int(10)	varchar (25)	varchar (25)	varchar (10)	varchar (25)	float(11,1)	float(10,2)

4.5 PatientTable:

patient id	patient name	admission date	admission time	address	Mobileno	mail	city	Pinc ode
int(10)	varcha r(50)	date	time	varchar(250)	varchar(15)	varchar(30)	varchar(25)	varchar(20)

Loginid	password	Bloodgroup	gender	dob	status
varchar(50)	varchar(25)	varchar(20)	varchar(10)	date	varchar(10)

4.6 PrescriptionTable:

Prescriptio n_id	Treatment _record_id	Doctorid	Patient id	Delivery _type	Delivery _id	Prescription _date	status	Appointment id
int(10)	int(10)	int(10)	int(10)	varchar (10)	int(10)	date	varchar (10)	int(10)

4.7 ServiceTypeTable:

service_type_id	service_type	Service charge	description	status
int(10)	varchar(100)	float(10,2)	text	varchar(10)

4.8 TreatmentTable:

treatmentid	treatmenttype	treatment_cost	note	status
int(10)	varchar(25)	decimal(10,2)	text	varchar(10)

4.9 UserTable:

userid	loginname	password	patientname	mobilenno	email	createddateandtime
int(11)	varchar(50)	varchar(10)	varchar(50)	varchar(15)	varchar(50)	datetime

5.

RESULT AND DISCUSSION

Since we are entering details of the patients electronically in the “Online Hospital Management System”, data will be secured. Using this application we can retrieve patient’s history with a single click. Thus processing information will be faster. It guarantees accurate maintenance of Patient details. It easily reduces the bookkeeping task and thus reduces the human effort and increases accuracy speed.

It is unreasonable to consider a computer based information system complete or finished; the system continues to evolve throughout its life cycle, even if it’s successful. It is the case with this system too. So, certain aspects of the system can be modified as operational experience is gained with it. We can focus on more efficient and streamlined processing system and better coordination between customers and the finance leading to desired results in terms of quality, service, profits, market, share and competitiveness. In future enhancements we can add a Chat Bot for support in the website 24x7. We can also convert this into an Android and iOS Application.

6. CONCLUSIONANDFUTUREENHANCEMENT

CONCLUSION

This project has been a rewarding experience in more than one way. Theentireprojectworkhasenlightenedusinthefollowingareas.We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation. Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed. Scheduling a project and adhering to that schedule creates a strong sense of time management. Sense of teamwork has developed and confidence of handling real life project has increased to a great extent. Initially, there were problem with the validation but with discussions, we were to implement validations.

FUTURE ENHANCEMENTS

- Diagnostics billings system.
- Chat box
- Online payment

REFERENCES

1. BeginningPHP5,Apache,andMySQLWebDevelopment(Programmertoprogrammer) by ElizabethNaramore.
2. F.Droma, Project report on “An automated System for patient Record System”,DepartmentofInformationTechnologymakerereuniversity.(2009).
3. Fujisoft(2012).FujisoftHospitalSolutions(Japan).Accessedfrom:http://www.fsi.co.jp/e/solutions/hospital_solutions/index.html, accessed 21 March2013).
4. PHP MySQL Website Programming: Problem - Design – Solution by Chris Lea,MikeBuzzard, DilipThomas ,JesseyWhite-Cinis.
5. HowtoDoEverythingwithPHPandMySQLbyVikramVaswani.
6. MySQL/PHPDatabaseApplications,2ndEditionbyBradBulger.

APPENDICIES

A.SOURCECODE

5.2Samplecode:

Index.php:

```
<?phpinclude("header.ph
p");
?>

<divstyle="background-color:#fff;"class="wrappercol1">

    <divid="container">

        <divid="content">

            <h1>WelcometoHospital Assist</h1>

            <p>The goal of the Hospital Assist is to improve the efficiency of the health
byreducingtheoveralltimeandcostusedtocreatedocumentsandretrieveinformation.
.</p>

            <p>The main feature of E-Knowledge in Doctor appointment system is to
providethebrowsertogetappointmentsfromadoctorthroughinternetinsteadofgoingthere
andfixing anappointment..</p>

            <p>    Everyone needs to have Medical attention at any time. So we
alloweveryuserto registerfreelyat any time..</p>

            <p>Doctor appointment System maintains patients prescriptions so that
theirmedicaldetailsarealwaysavailableinInternet,whichwillbemoreconvenientforthepatie
nts.This will be more comfortable for thepatient..</p>

            <divclass="homecontent">

                <ul>

                    <li>

                        <h2>BookyourAppointmentthroughOnline</h2>

                        <p><a href="patientappointment.php"><imgsrc="images/appointmentimg.jpg"a
lt=""style="width:300px;height:200px;"/></a></p>

                    </li>
```

```

        <liclass="last">

            <h2>LoginPanelforExistingUsers</h2>

            <p><a href="patientlogin.php"></a></p>

            </li>

            <li>

                <br><br>

                <h2>LoginPanelforAdmin</h2>

                <p><a href="adminlogin.php"></a></p>

                </li>

            <liclass="last">

                <br><br>

                <h2>LoginPanelforDoctor</h2>

                <p><a href="doctorlogin.php"></a></p>

            </li>

        </ul>

        <div class="clear"></div>

    </div>

</div>

<div class="clear"></div>

</div>

<div>

    <?php include("footer.php");

?>

patientappointment.php

```

<?php

```

session_start();include("header.php")
;include("dbconnection.php");if(isset(
$_POST[submit]))
{
    if(isset($_SESSION[patientid]))
    {
        $lastinsid=$_SESSION[patientid];
    }
    else
    {
        $dt=date("Y-m-d");
        $tim=date("H:i:s");

        $sql="INSERT INTO
patient(patientname,admissiondate,admissiontime,address,city,mobileno,mail,loginid,
password,gender,dob,status)values('$_POST[paciente]','$dt','$tim','$_POST[textarea]'
,'$_POST[city]','$_POST[mobileno]','$_POST[mail]','$_POST[loginid]','$_POST[passw
ord]','$_POST[select6]','$_POST[dob]','Pending')";

        if($qsql=mysqli_query($con,$sql))
        {
            /* echo "<script>alert('patient record
insertedsuccessfully...');</script>";*/
        }
        else
        {
            echomysqli_error($con);
        }
        $lastinsid=mysqli_insert_id($con);
    }
}

```

```

        $sqlappointment="SELECT * FROM appointment
WHEREappointmentdate='$_POST[appointmentdate]'
ANDappointmenttime='$_POST[appointmenttime]' AND doctorid='$_POST[doct]'
ANDstatus='Approved'";

        $qsqlappointment=mysqli_query($con,$sqlappointment);if(m
ysqli_num_rows($qsqlappointment) >=1)
        {
            echo"<script>alert('Appointmentalreadyscheduledforthistime..');</script>
";
        }
        else
        {
            $sql="INSERT INTO
appointment(appointmenttype,patientid,appointmentdate,appointmenttime,app_reason,status,d
epartmentid,doctorid)values('ONLINE','$lastinsid','$_POST[appointmentdate]','$_POST[appoint
menttime]','
$_POST[app_reason]','Pending','$_POST[department]','$_POST[doct]')";
            if($qsql=mysqli_query($con,$sql))
            {
                echo"<script>alert('Appointmentrecordinsertedsuccessfully...');</s
cript>";
            }
            else
            {
                echomysqli_error($con);
            }
        }
    }
    if(isset($_GET[editid]))
    {
        $sql="SELECT*FROMappointmentWHEREappointmentid=$_GET[editid]";

```

```

        $qsql=mysqli_query($con,$sql);
        $rsedit=mysqli_fetch_array($qsql);
    }
    if(isset($_SESSION[patientid]))
    {
        $sqlpatient="SELECT*FROMpatientWHEREpatientid='".$_SESSION[patientid]";
        $qsqlpatient=mysqli_query($con,$sqlpatient);
        $rspatient=mysqli_fetch_array($qsqlpatient);
        $readonly=" readonly";
    }
    ?>
<divclass="wrapper col2">
    <divid="breadcrumb">
        <ul>
            <liclass="first">AddNewAppointment</li></ul>
        </div>
    </div>
</div>
<divclass="wrappercol4">
    <divid="container">
        <?phpif(isset($_POST[s
ubmit]))
        {
            if(mysqli_num_rows($qsqlappointment)>=1)
            {
                echo "<h2>Appointment already scheduled for ". date("d-M-
Y",strtotime($_POST[appointmentdate])) . " " . date("H:i
A",strtotime($_POST[appointmenttime])). "</h2>";
            }

```

```

else
{
    if(isset($_SESSION[patientid]))
    {
        echo"<h2>Appointmenttakensuccessfully..</h2>";
        echo"<p>Appointmentrecordisinpendingprocess.Kinldyhec
kthe appointment status. </p>";
        echo"<p><a href='viewappointment.php'>ViewAppointment
record</a>.</p>";
    }
    else
    {
        echo"<h2>Appointmenttakensuccessfully..</h2>";
        echo"<p>Appointmentrecordisinpendingprocess.Pleasewait
forconfirmationmessage..</p>";
        echo"<p><a href='patientlogin.php'>ClickheretoLogin</a>
</p>";
    }
}
}
else
{
    ?>
    <form method="post" action="" name="frmpatapp" onsubmit="return validateform
()">
    <table width="532" border="3">
    <tbody>
    <tr>
    <td width="34%">PatientName</td>

```

```
<td width="66%"><input type="text" name="patiente"
id="patiente" value="<?php echo $rspatient[patientname];?>"<?php echo $readonly;?>><
/td>
```

```
</tr>
```

```
<tr>
```

```
<td height="62">Address</td>
```

```
<td><textarea name="textarea" id="textarea" <?php echo $readonly;?>><?php echo $rspatient[address];?></textarea></td>
```

```
</tr>
```

```
<tr>
```

```
<td>City</td>
```

```
<td><input type="text" name="city" id="city" value="<?php echo $rspatient[city];
?>"<?php echo $readonly;?>></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Mobile Number</td>
```

```
<td><input type="text" name="mobileno" id="mobileno" value="<?php echo $rspatient[mobileno];?>"<?php echo $readonly;?>></td>
```

```
</tr>
```

```
<tr>
```

```
<td>E-mail</td>
```

```
<td><input type="mail" name="email" id="email" value="<?php echo $rspatient[mail];?>"<?php echo $readonly;?>></td>
```

```
</tr>
```

```
<?php
```

```
if(!isset($_SESSION[patientid]))
```

```
{
```

```
?>
```

```
<tr>
```

```

        <td>Login-ID</td>

        <td><input type="text" name="loginid" id="loginid" value="<?php echo
$rspatient[loginid];?>"<?php echo $readonly;?>></td>

    </tr>

    <tr>

        <td>Password</td>

        <td><input type="password" name="password" id="password" value="<?php echo
$rspatient[patientname];?>"<?php echo $readonly;?>></td>

    </tr>

<?php
    }
?>

    <tr>

        <td>Gender</td>

        <td>

            <?php
                if(isset($_SESSION[patientid]))
                {
                    echo $rspatient[gender];
                }
                else
                {
                    ?>

                    <select name="select6" id="select6">
                        <option value="">Select</option>

                        <?php
                            $arr=array("Male","Female");
                            foreach($arr as $val)

```

```

        {
            echo"<optionvalue='$val'>$val</option>";
        }
    ?>
</select>

        <?php
        }
        ?>

</td>

</tr>

<tr>

    <td>DOB</td>

    <td><inputtype="date"name="dob"id="dob"value="<?phppecho
$rspatient[dob];?>"<?phppecho$readonly;?>></td>

</tr>

<tr>

    <td><strong>EnterAppointmentDate</strong></td>

    <td><inputtype="date"min="<?phppecho date("Y-m-
d");?>"name="appointmentdate"id="appointmentdate"></td>

</tr>

<tr>

    <td><strong>EnterAppointmentTime</strong></td>

    <td><inputtype="time"name="appointmenttime"id="appointmenttime"></td>

</tr>

<tr>

    <td><strong>Department</strong></td>

    <td>

```

```

        <selectname="department"id="department"onchange="loaddoctor(this.value)">
        <optionvalue="">Selectdepartment</option>
    <?php
        $sqldept="SELECT*FROMdepartmentWHERE
status='Active'";
        $qsqldept =
        mysqli_query($con,$sqldept);while($rsdept=mysqli_fetch_array($qsqldept))
        {
            echo"<option
value='$rsdept[departmentid]'>$rsdept[departmentname]</option>";
        }
    ?>
</select>
</td>
</tr>

    <tr>
        <td><strong>Doctors</strong></td>
        <td>
            <div id="divdoc">
                <selectname="doct"id="doct">
                <optionvalue="">Selectdoctor</option>
                </select>
            </div>
        </td>
    </tr>

```

B.SCREENSHOTS

6.1 HomePage:



Fig1

This is the home page of our project. In this page a new user i.e., a patient or a doctor can sign-up. Any existing user can login with their proper username and password.

6.2 OnlineAppointmentPage:

A screenshot of the 'Online Appointment' page in the 'Online Hospital Management System'. The browser's address bar shows 'localhost/Online-Hospital-Management-System/patientappointment.php'. The page has the same header and navigation menu as the home page. Below the navigation menu, there is a section titled 'Add New Appointment'. This section contains a form with the following fields: 'Patient Name', 'Address', 'City', 'Mobile Number', 'E-mail', 'Login-ID', 'Password', 'Gender' (a dropdown menu with 'Select' as the current selection), 'DOB' (a date picker showing 'dd-mm-yyyy'), and 'Enter Appointment Date' (a date picker showing 'dd-mm-yyyy').

Fig2

This is the appointment registration page where a patients who wish to visit the doctor can book an appointment by filling up the required details. After registration, most of the details like name and address provided during first sign up are auto filled into the form. Some additional details like date and time of appointment have to be given and previous details also can be modified.

6.3 PatientRegistrationPage:

The screenshot shows a web browser window with the URL `localhost/Online-Hospital-Management-System/patient.php`. The page features a header with the logo "ONLINE HOSPITAL MANAGEMENT SYSTEM" and a navigation menu with links: Home, About US, Online Appointment, Login, Registration, and Contact US. Below the navigation menu is a button labeled "Add New Patient". The main content area is titled "Patient Profile Registration Panel" and contains a form with the following fields: Patient Name, Address, E-mail, Mobile Number, City, PIN Code, Login ID, and Password. Each field has a corresponding input box.

Fig3

This is the registration page which is only for new users/patients who can register by filling up the necessary form details.

6.4 PatientLoginPage:

The screenshot shows a web browser window with the URL `localhost/Online-Hospital-Management-System/patientlogin.php`. The page features the "ONLINE HOSPITAL MANAGEMENT SYSTEM" logo at the top. Below the logo is a navigation bar with links: Home, About US, Online Appointment, Login, Registration, and Contact US. The main heading is "Patient Login panel". The instructions state: "Kindly login to the website by entering Login ID and Password..". The login form includes two input fields for "Login ID" and "Password", a "Login" button, and a link for "New user Click Here to Register | Forgot Password". The footer contains the text "Copyright © ,2019 - All Rights Reserved".

Fig4

6.5 AdminLoginPage:

The screenshot shows a web browser window with the URL `localhost/Online-Hospital-Management-System/adminlogin.php`. The page features the "ONLINE HOSPITAL MANAGEMENT SYSTEM" logo at the top. Below the logo is a navigation bar with links: Home, About US, Online Appointment, Login, Registration, and Contact US. The main heading is "Admin Login Panel". The instructions state: "Kindly enter Login ID and Password". The login form includes two input fields for "Login ID" and "Password", a "Login" button, and a link for "New user Click Here to Register | Forgot Password". The footer contains the text "Copyright © ,2019 - All Rights Reserved".

Fig5

This is the admin login page designed for the admin to login using proper credentials.

6.6 DoctorLoginPage:

The screenshot shows a web browser window with the title 'Online Hospital Management System'. The address bar shows the URL 'localhost/Online-Hospital-Management-System/doctorlogin.php'. The page features a large blue header with a white cross icon and the text 'ONLINE HOSPITAL MANAGEMENT SYSTEM'. Below the header is a navigation bar with links: Home, About US, Online Appointment, Login, Registration, and Contact US. The main content area is titled 'Doctor Login Panel' and contains the instruction 'Kindly Enter Login ID and Password..'. There are two input fields: 'Login ID' and 'Password'. A blue 'Submit' button is located below the input fields. The footer is a blue bar with the text 'Copyright © ,2019 - All Rights Reserved'.

Fig6

This is the doctor login page designed for the registered doctor to login with the credentials they have registered with.

