HOSPITAL MANAGEMENT SYSTEM

Project Report Submitted By

FRANK MATHEWS THOMAS

Reg. No.: AJC17MCA-I018

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

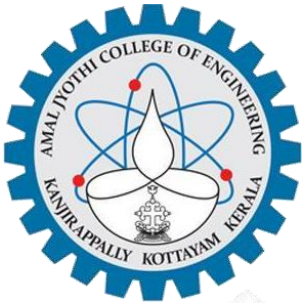
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**HOSPITAL MANAGEMENT SYSTEM"** is the bonafide work of **FRANK MATHEWS THOMAS (Reg.No: AJC17MCA-I018)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Anit James
Internal Guide

Rev. Fr. Dr. Rubin Thottupurathu Jose
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"HOSPITAL MANAGEMENT SYSTEM"** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **FRANK MATHEWS THOMAS**

KANJIRAPPALLY **Reg. No: AJC17MCA-I018**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Rev. Fr. Dr. Rubin Thottupurathu Jose and Ms.Meera Rose Mathew** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Ms. Anit James** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

FRANK MATHEWS THOMAS

ABSTRACT

The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient’s information, Appointment details, Pharmacy, video consultation via online etc.

The project ‘Hospital Management System’ is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using software which is one of the best and the easiest software to keep our information

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement.

Entire application is web based and built on 3 tier architecture using the latest technologies. The database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Doctor, Admin, Pharmacy, Patient appointment ,video consultation etc.

CONTENT

List of Abbreviation

IDE	Integrated Development Environment
HTML	Hyper Text Markup Language.
CSS	Cascading Style Sheet
SQL	Structured Query Language
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and video consultation. The software has the facility to give a unique detail for every patient and stores the details of every patient and the staff automatically.

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

1.2 PROJECT SPECIFICATION

The proposed system is made to help the customers for an easy and convenient way of booking an appointment and also helps easy to their work and source. We will also provide users to give feedbacks, they can view the appointments details, payment details, complaint details, pharmacy details etc.

The system includes 5 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add or update doctor categories, patients and appointment details etc. Admin can view all the registered doctor and appointments and also can able to approve or reject users and also can able to view all registered customer details.

2. Patient Module

Customer can register and book appointments they can upload their details to hospital and do secure online payment. Customer can also find doctors nearby them and can add feedbacks and complaints to them.

3. Doctor Module

Doctor can view registered appointments and specialization doctors ,also change the session logs of patients and session logs of doctors.

4. Video Consultation Module

Video Consultation is a video communication web application designed for online video consultations, interactive live presentations, trainings, webinars, coaching and online collaboration. This Video Consultation edition integrates into Moodle as an activity module so video consultation seminars can be added to courses as activities.

5. Pharmacy Module

The Pharmacy Module allows users to monitor medication stock, invoice funding, and pharmacy supplier. Users are able to track patient-specific prescriptions and generate reports for clinic level inventory tasks such as drug stock refills and the quantity of each drug dispensed.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analysed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analysing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM OF PROJECT MANAGEMENT SYSTEM

Existing system is not a fully automated system. Customer can register and book the appointments. Each customer can create their own profile .The proposed system rectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient in managing project details including appointment approval, less transparency, no standardized packages.
- Project often delayed with no progress visibility.
- Human effort is needed
- Customers do not get a chance to identify the quality of specificality used

2.4 PROPOSED SYSTEM

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

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

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

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

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

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

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

8. **Reduces redundancy:** The most important benefit of this system is that it reduces the redundancy of data within the data.
9. Work load: Reduces the work load of the data store by helping in easy updates of the products and providing them with the necessary details together with financial transactions management.
10. Easy statements: Month-end and day-end statement easily taken out without getting headaches on browsing through the day end statements.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- **You are able to get creative and innovative project plans:-**
  Our customers will get creative project plans with affordable oppurtunity and they have the option to view profile and can send feedbacks. Also can able to do proper management and can also improve and increase their work and income source.
- **Better security: -**
  For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.
- **Ensure data accuracy: -**
  The proposed system eliminates the manual errors while entering the details of the users during the registration.
- **Better service: -**
  The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, Hospital Management System was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input,

output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. Hospital Management System, GUI is simple so that users can easily use it.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 GB

Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML, CSS, SCSS

Backend - MySQL, PHP

Client on PC - Windows 10 and above.

Technologies used - JS, HTML5, AJAX, jQuery, PHP, CSS, SCSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

MySQL is a database management system.

database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database

management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational.

relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a languagespecific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual,

"SQL92" refers to the standard released in 1992,

"SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

The MySQL Database Server is very fast, reliable, scalable, and easy to use.

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML stands for Unified Modelling Language. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modelling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design.

4.2.1 Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are

employed in UML (Unified Modelling Language), a standard notation forth modelling of real-world objects and systems

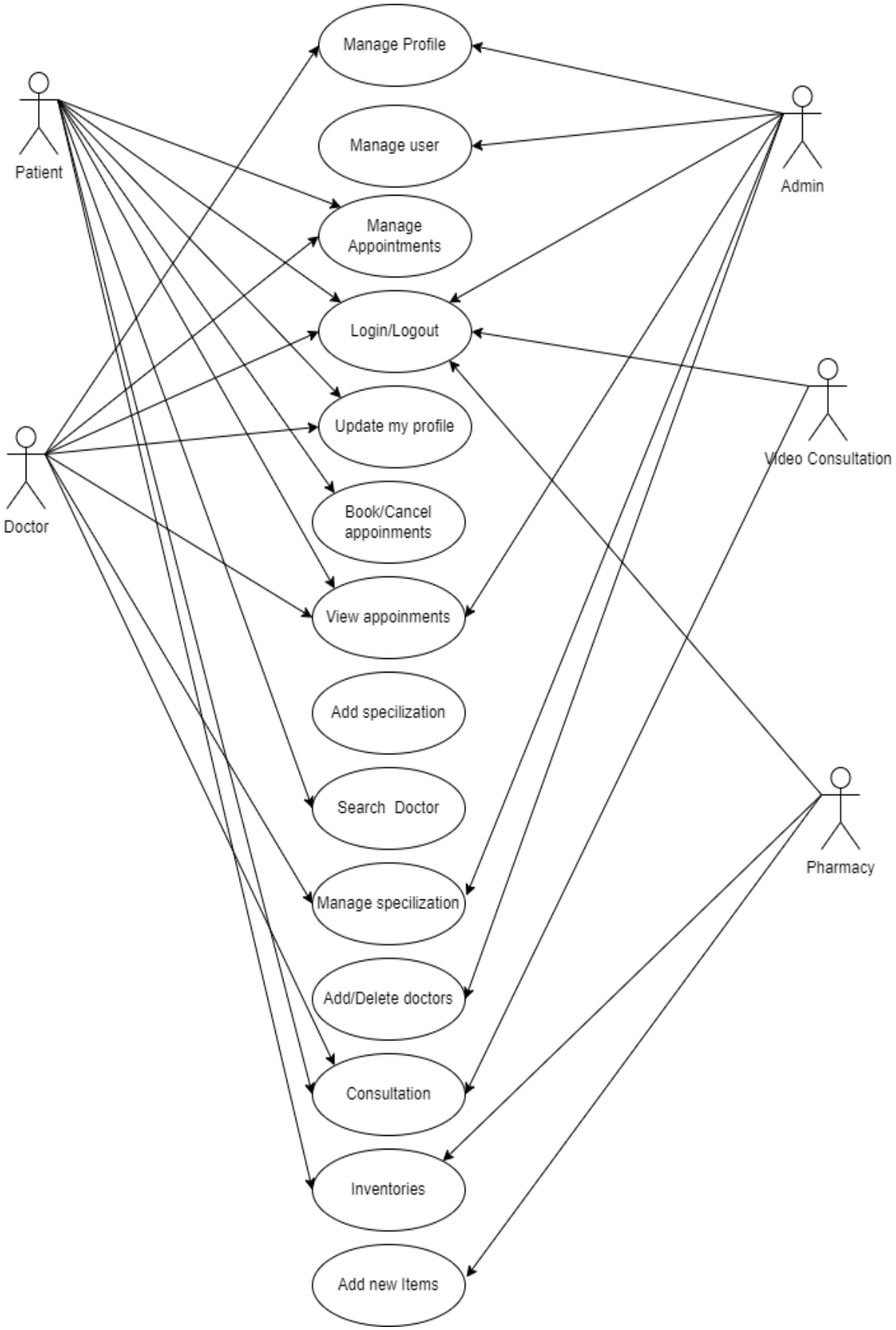
A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1: Use case diagram for Hospital Management System



4.2.2 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

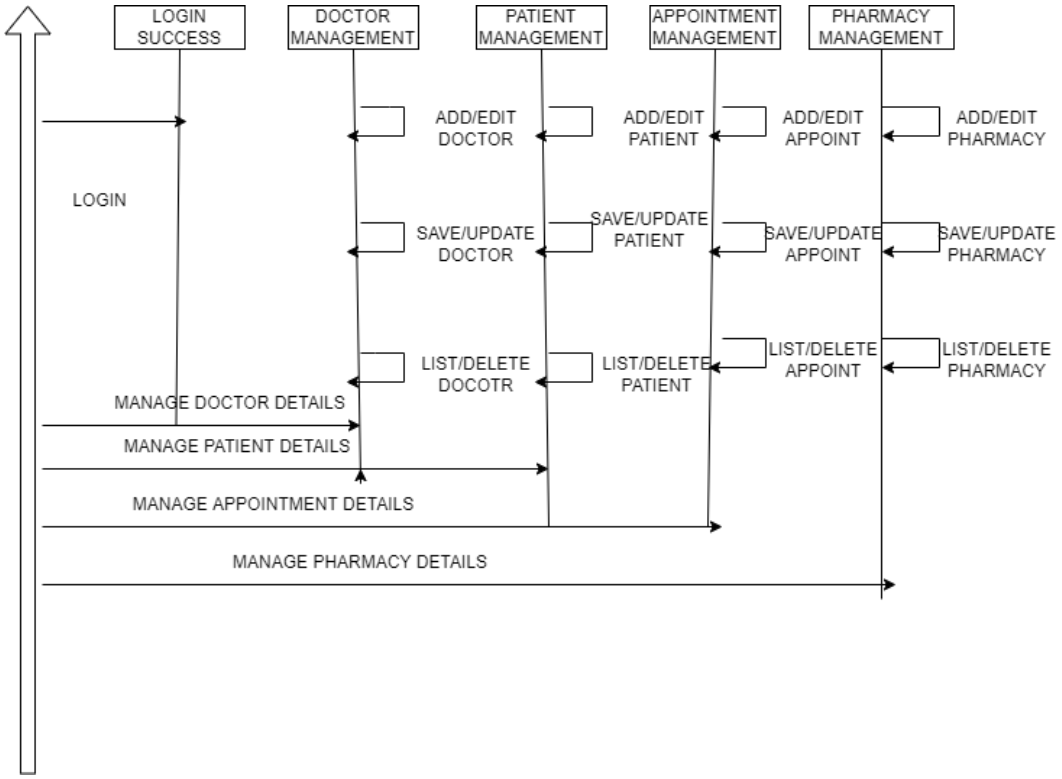
iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

Fig 2: Sequence diagram for Hospital Management System



4.2.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

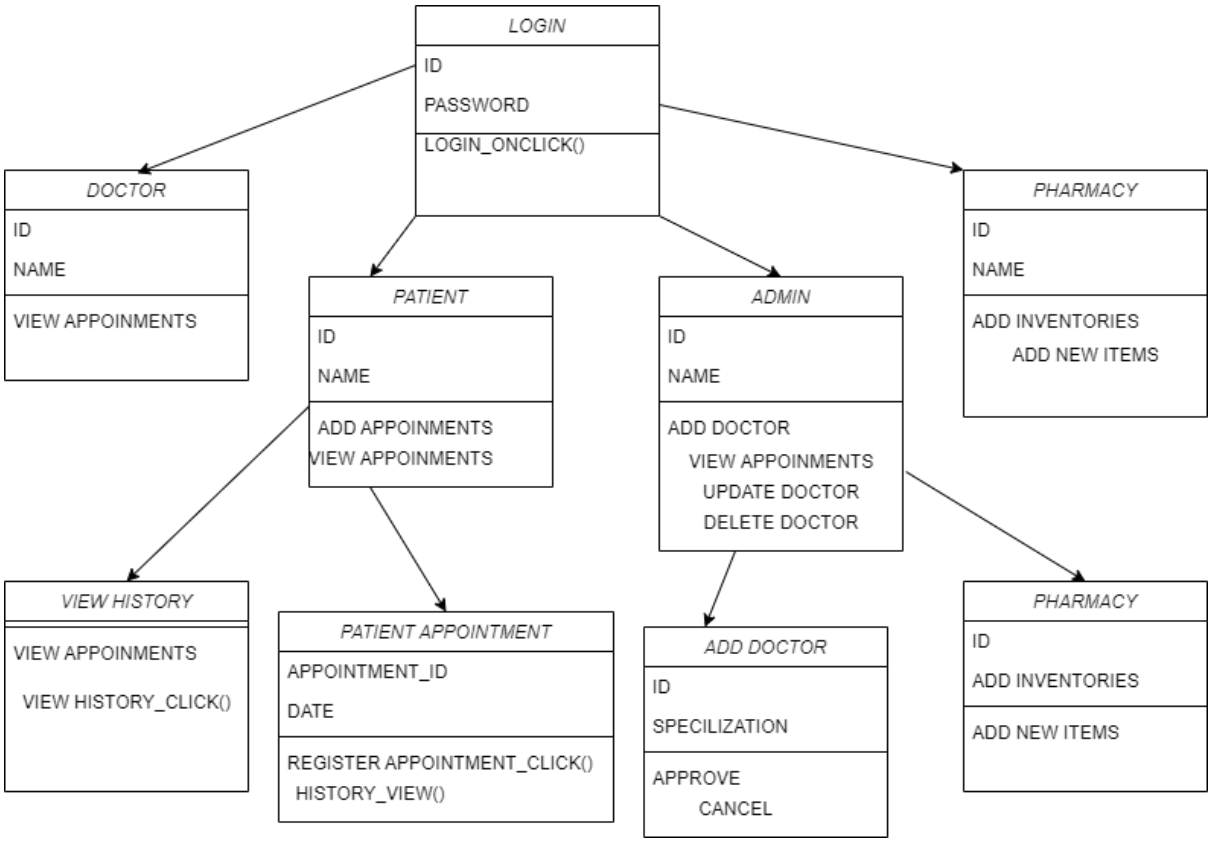
Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The following points should be remembered while drawing a class diagram −

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.

- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing, it should be understandable to the developer/coder.

Fig 3: Class diagram for Project Management System

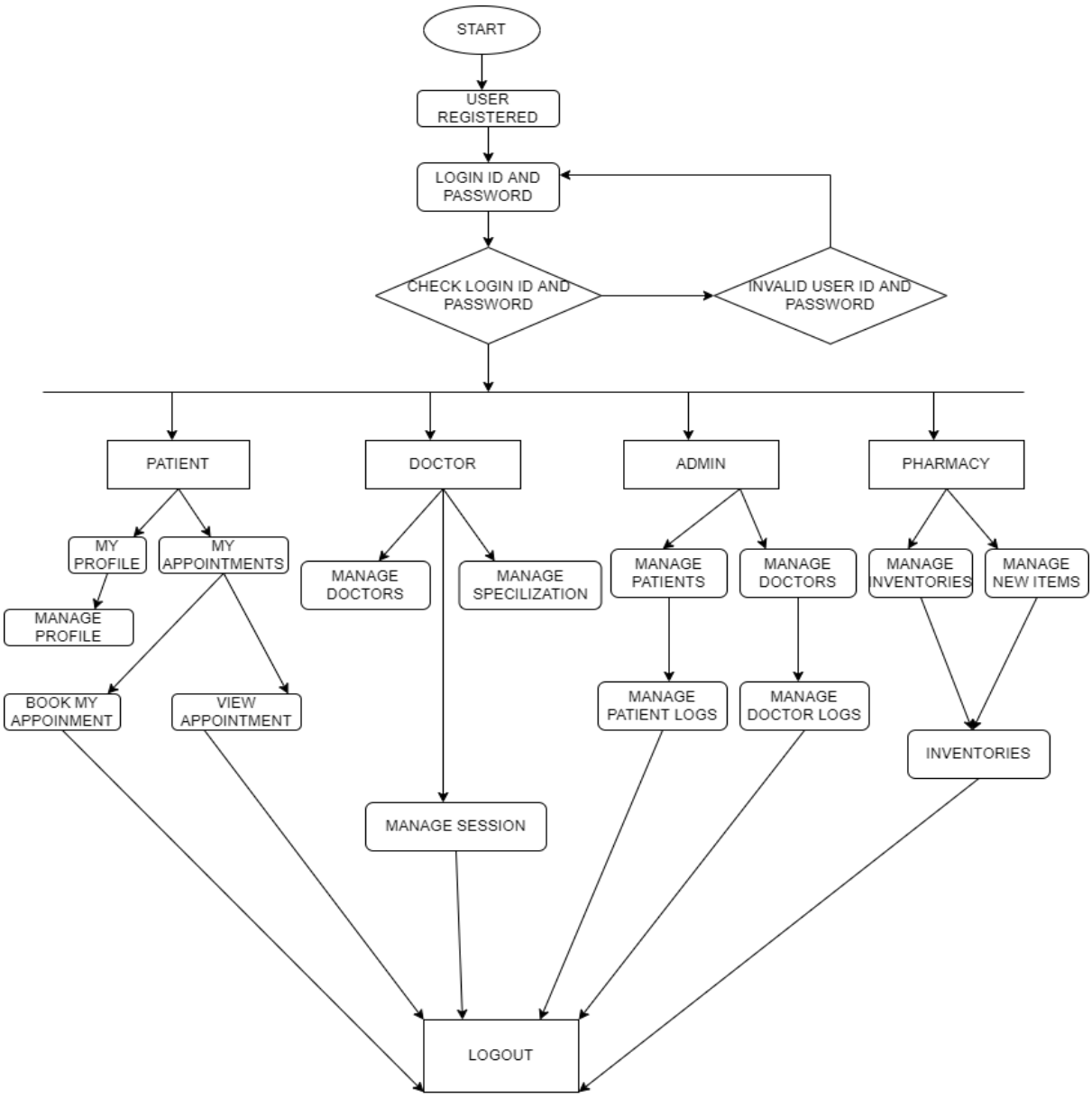


4.2.4 Activity Diagram

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

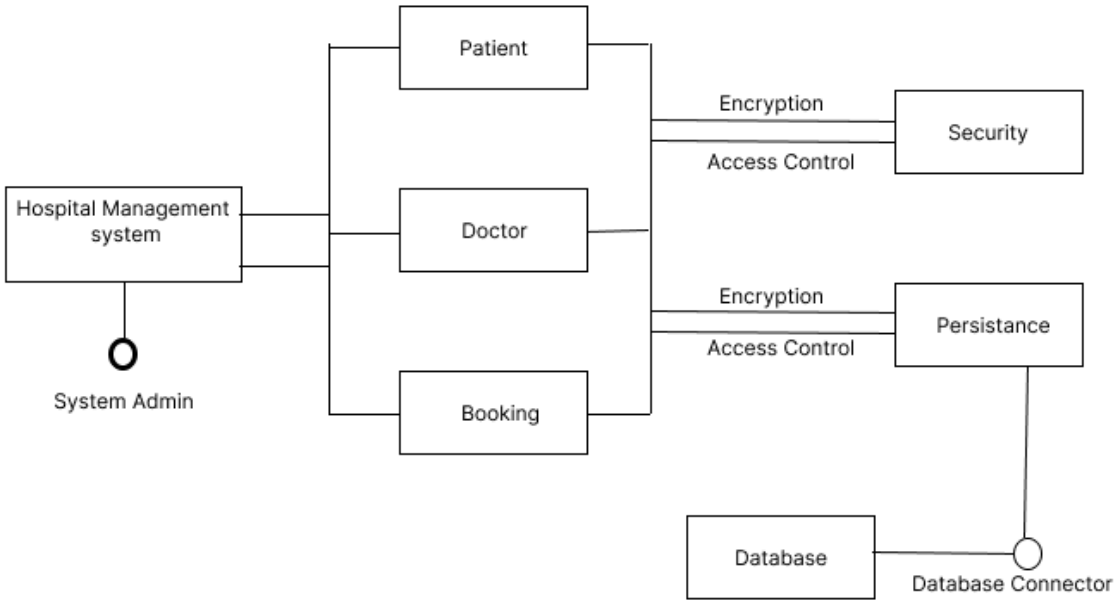
An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

Fig 4: Activity diagram for Project Management System



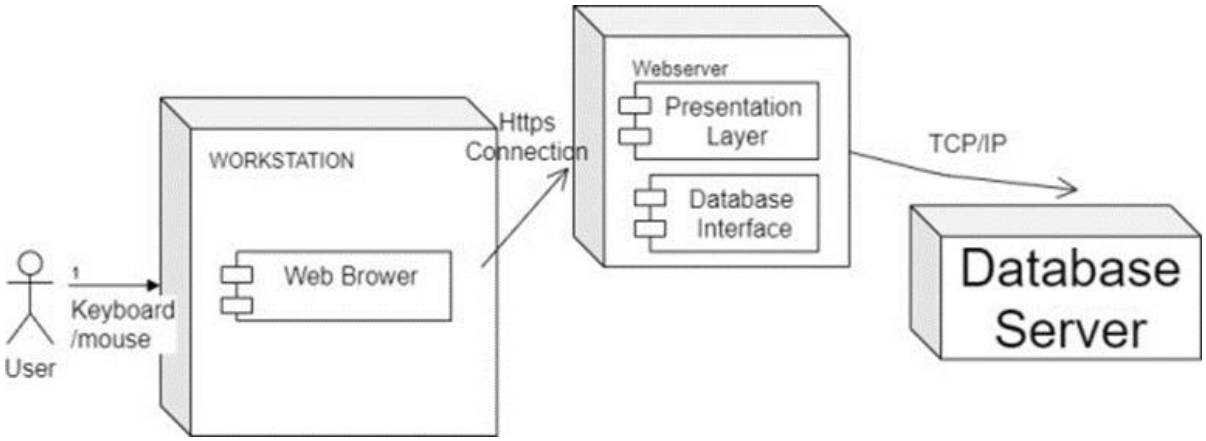
4.2.5 COMPONENT DIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviour is explained by the provided and required interfaces



4.2.6 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.



4.2.7 Deployment Diagram

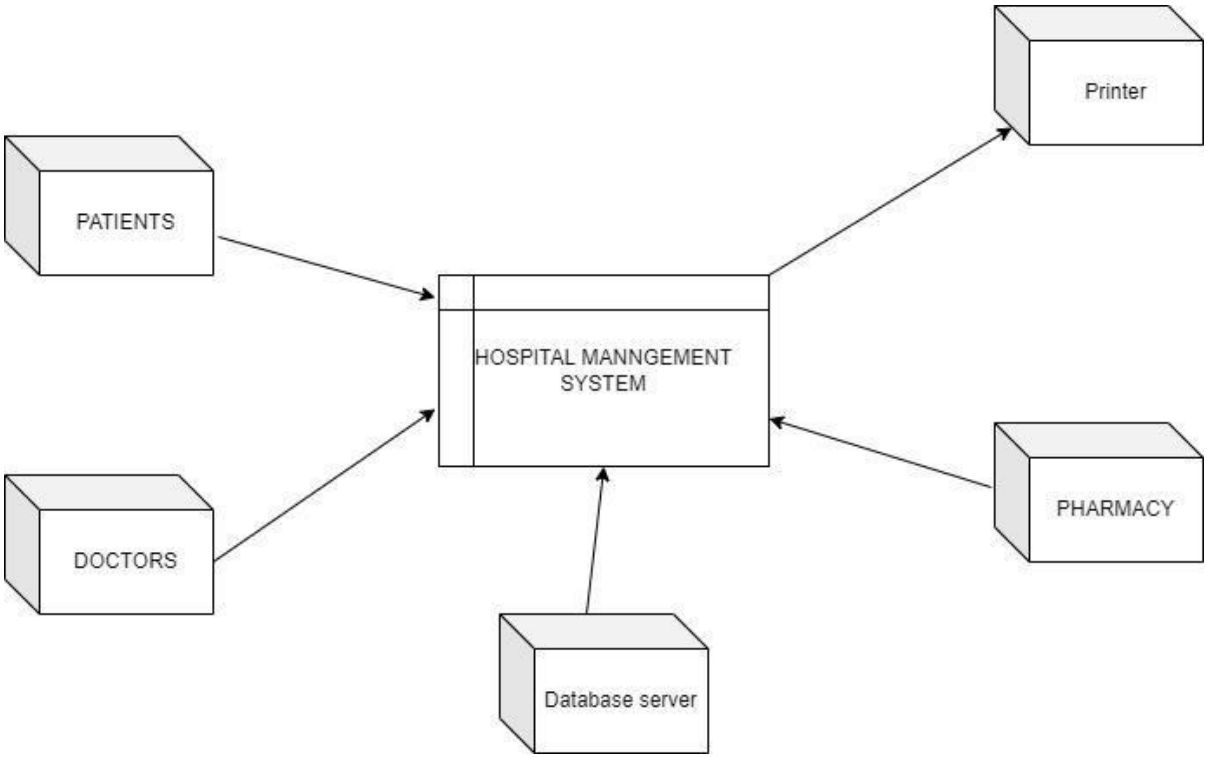
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The purpose of deployment diagrams can be described as −

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

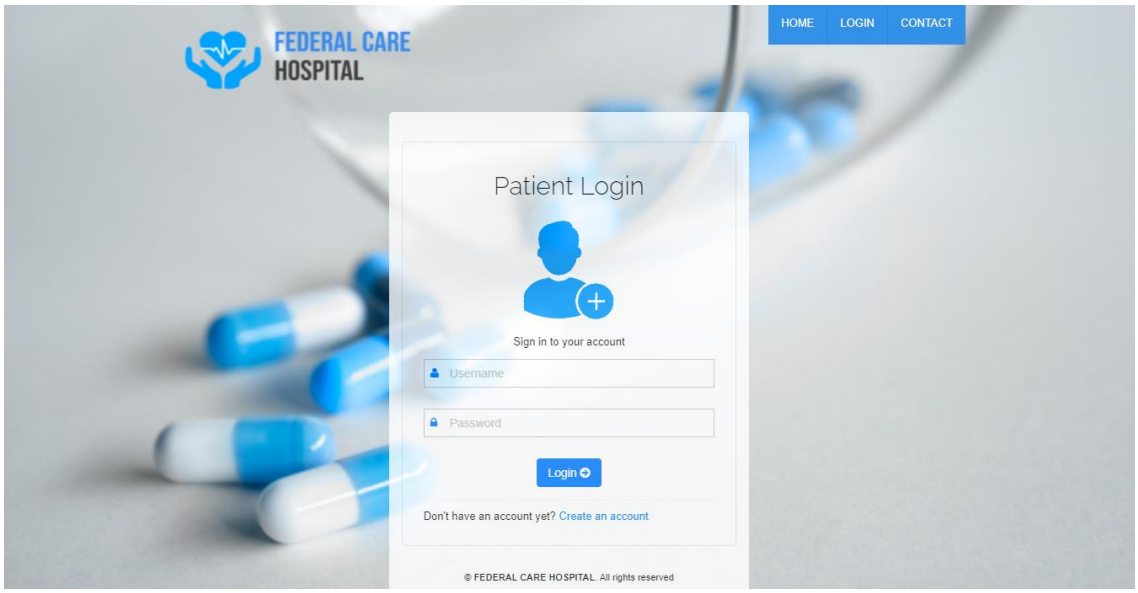
Fig 5: Deployment diagram for Project Management System



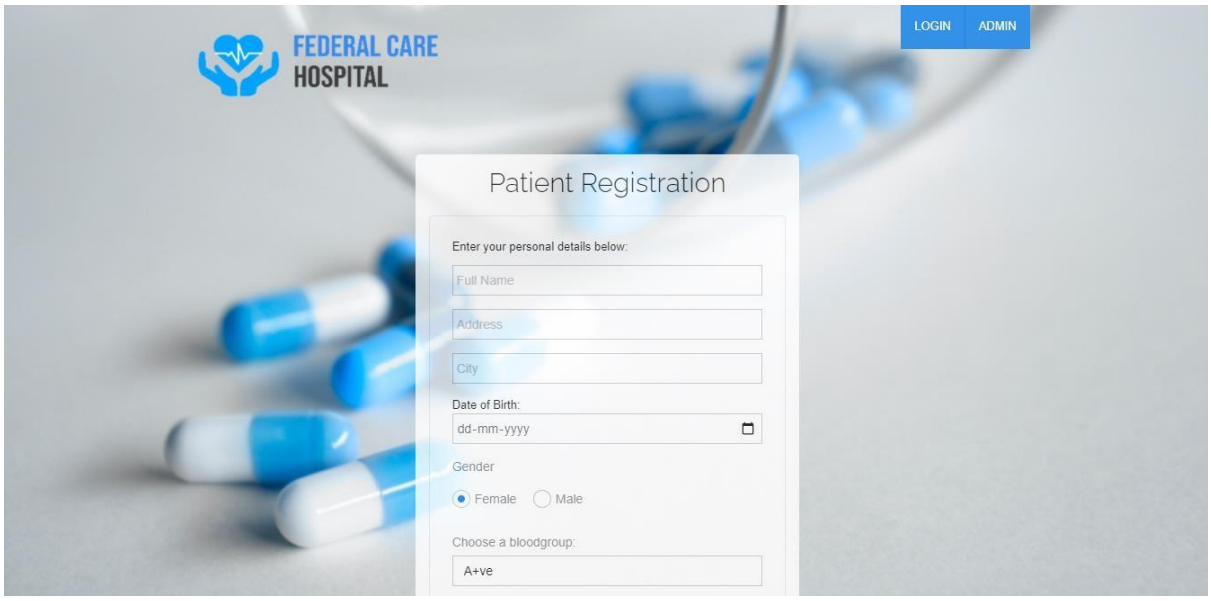
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1 Design Prototype

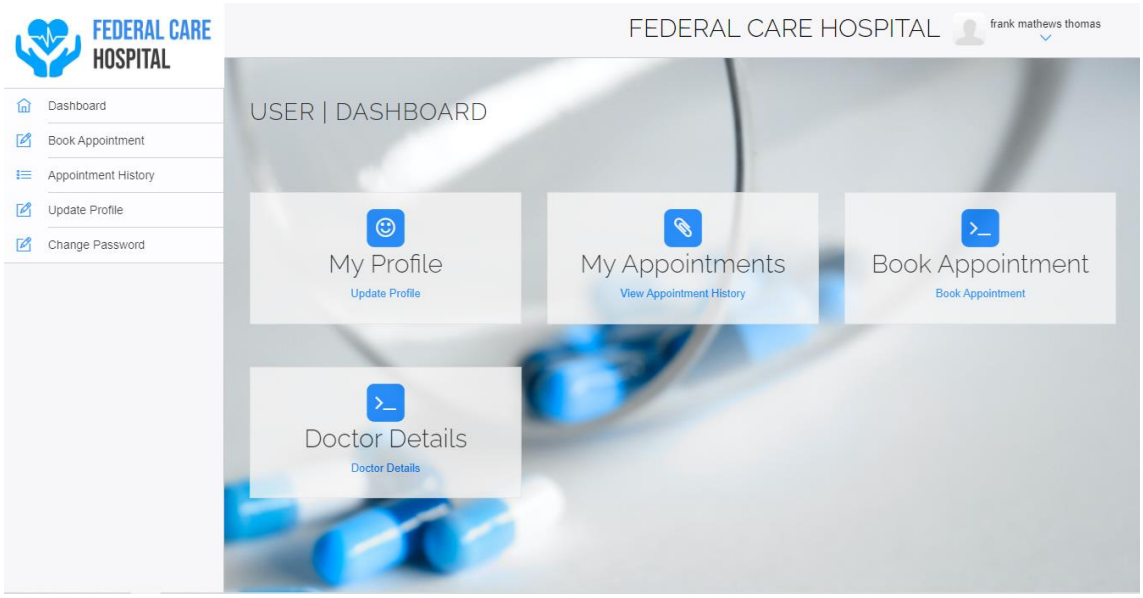
Login Page



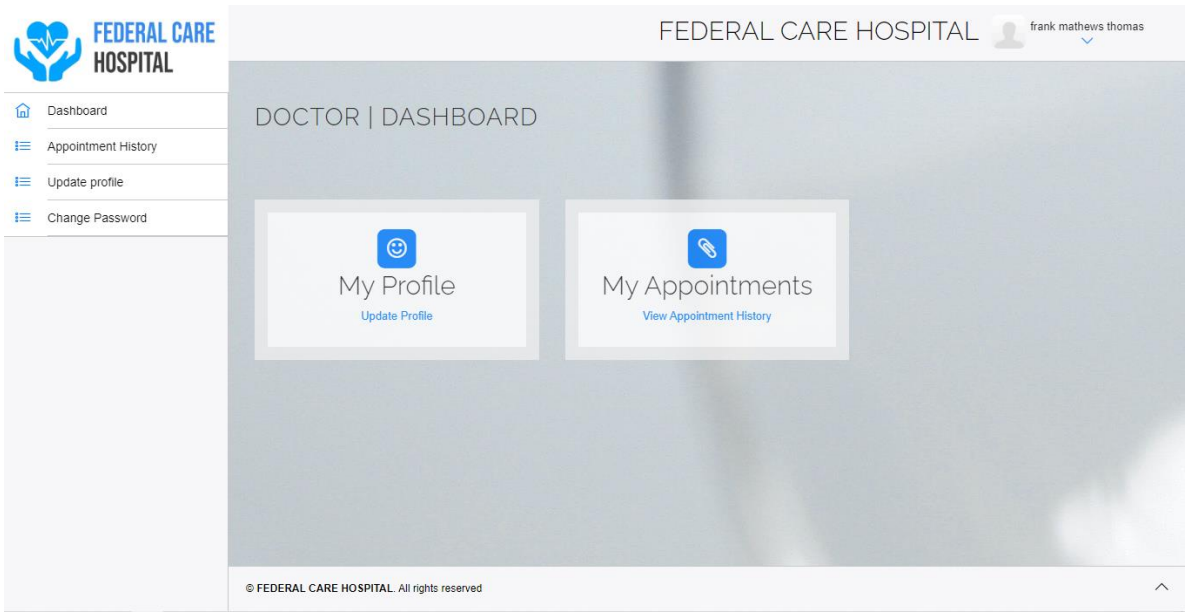
Registration Pages



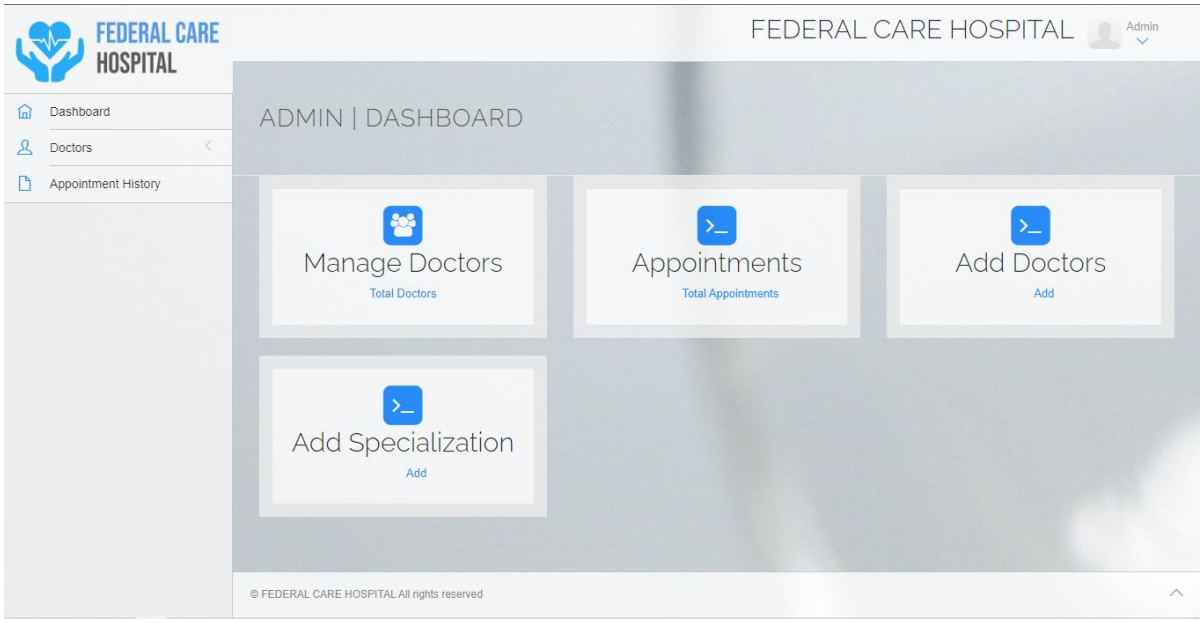
User Landing Page



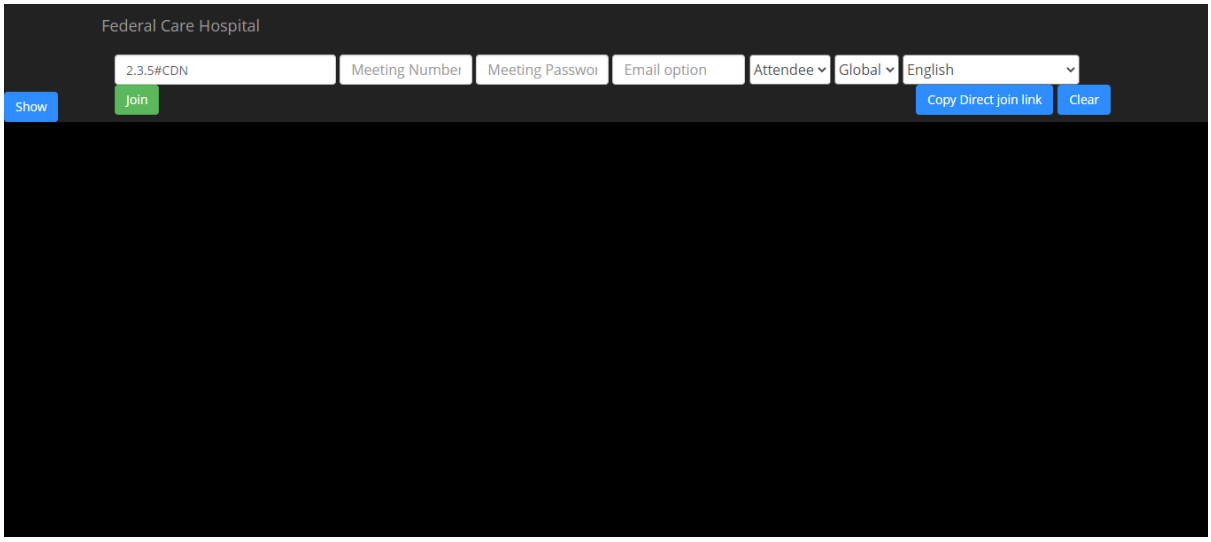
Docotor Landing page



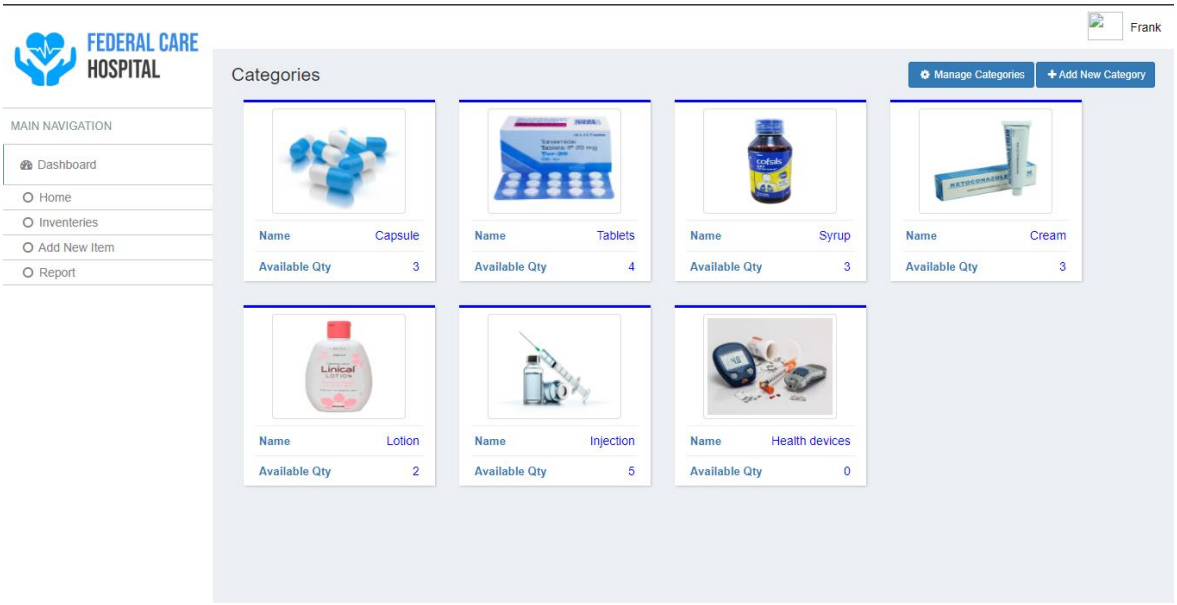
Admin Landing page



Video Consultation



Pharmacy Landing Page



4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key is Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor

by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute

Table Design

Table no-1

Table Name-tbl_register

Primary key-user_id

Table desc-Registration form for the patients

Field Name	Type	Constraints	Description
user_id	int	primary key	Login id
username	varchar		Username
address	varchar		Address
city	varchar		Current city
email	varchar		Email
password	varchar		Password
gender	varchar		Gender
dob	varchar		Date of birth
bloodgrp	varchar		Blood group
User_status	varchar		As defined 1

Table no-2

Table Name-Doctor

Primary key-user_id

Foreign key-User_id reference from Tbl_user table

Table desc-form for doctor desc

Field Name	**Type**	**Constraints**	**Description**
d_id	int	primary key	Doctor id
specilization	varchar		Specialization
User_id	int	foreign key	Login id
descrpt	varchar		Doctor desc
docFees	varchar		Doctor fees
contactno	varchar		Contact no
Year of Experience	varchar		Year of exp

Table no-3

Table Name-Doctor Specilization

Primary key-S_id

Table desc-Details about Doctor specilization

Field Name	**Type**	**Constraints**	**Description**
S_id	int	primary key	Specilization id
specilization	varchar		Specialization
creationDate	timestamp		Creation date

Table no-4

Table Name-Admin

Primary key-A_id

Table desc-Login details for Admin

Field Name	Type	Constraints	Description
A_id	int	primary key	Admin id
username	varchar		Username
password	varchar		Password
Status	varchar		As defined

Table no-5

Table Name-appoinment

Primary key-A_id

Foreign key-User_id reference from tbl_user

Table desc-Appoinment form for patients

Field Name	Type	Constraints	Description
A_id	int	primary key	Appointment id
User_id	int	foreign key	Login id
appointmentDate	varchar		Appointment date
appointmentTime	varchar		Appointment time
appointment status	varchar		Status
TokenNo	varchar		Token no
medicine	varchar		Medicine
doctorStatus	int		status

Table Name- no-6

Table Name-Prescription

Primary key-P_id

Foreign key-A_id refrence from Appoinment

Table desc- Prescription form patients

Field Name	Type	Constraints	Description
P_id	int	primary key	Prescription id
A_id	varchar	foreign key	Appointment id
Prescribe medicine	varchar		Medicine
Status	varchar		As defined

Table no-7

Table-Name-Complaint

Primary key-C_id

Foreign key-User_id reference from tbl_user

Table desc- Complaint form for users

Field Name	Type	Constraints	Description
C_id	int	primary key	Complaint id
User_id	int	foreign key	Login id
complaint	varchar		Complaint
delete	varchar		Delete complaint
status	varchar		Status
reply	varchar		reply

Table no-8

Table-Name-Pharmacy Category

Primary key-C_id

Table desc- Category management for pharmacy

Field Name	Type	Constraints	Description
C_id	int	primary key	Category id
Name	int	foreign key	Name
pic	varchar		Pictures
Details	varchar		details

Table no-9

Table -Inventory Category

Primary key-C_id

Table desc- Inventory management for pharmacy

Field Name	Type	Constraints	Description
id	int	primary key	Inventory id
cat_id	int	foreign key	Category id
supplier	varchar		Supplier
name	varchar		Name
unit	varchar		Unit
price	varchar		Price
description	varchar		Description
company	int		company

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as its vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appears to have been met.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers are always responsible for testing the individual units of the programs,

ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run

individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs
- Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested

using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Testing using Selenium

Selenium is one of the most widely used open-source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven, Jenkins, & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG, & JUnit for managing test cases and generating reports.

Test Case 1

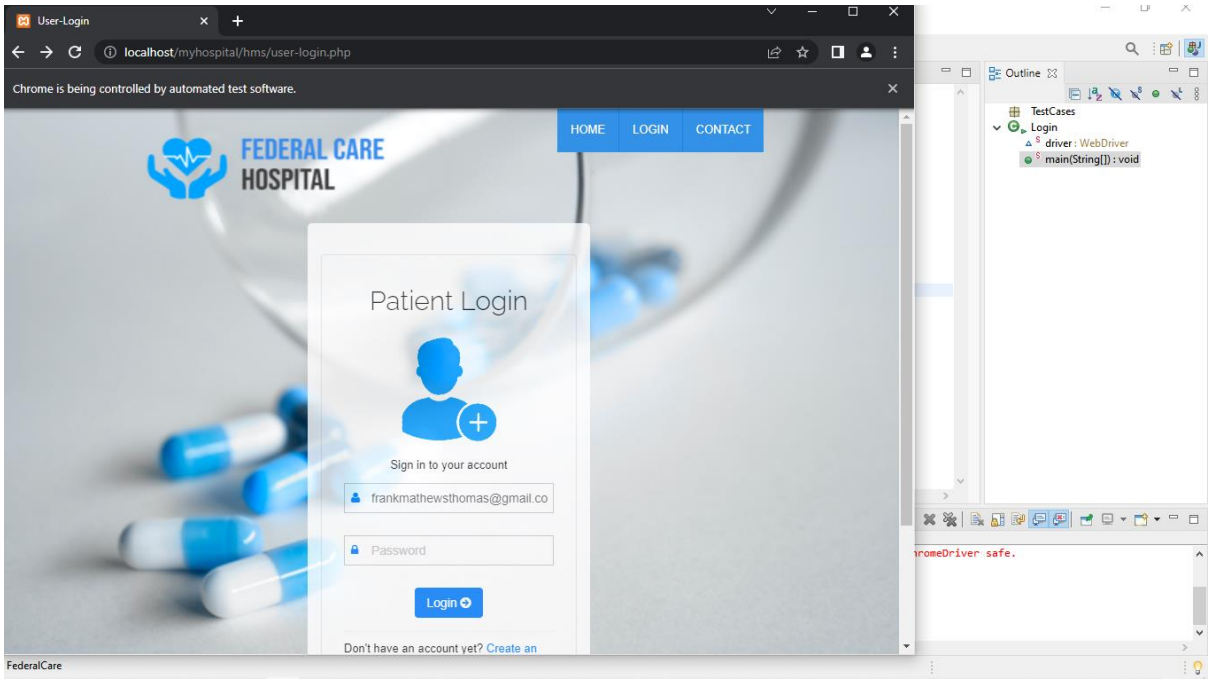
Project Name: Hospital Management System					
Login Test Case					
Test Case ID:login			**Test Designed By: Frank Mathews Thomas**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 24-05-2022		
Module Name:Login Screen					
Test Title : Verify login with valid username and password			**Test Execution Date:** 24-05-2022		
Description: Test the Login Page					
Pre-Condition : User has valid user name and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to LoginPage		Login Page shouldbe	Login page displayed	Pass
2	Provide Valid User name	User Name: frank@gmail.com	User should be ableto Login	User Logged in and navigated to Patient Dashboard with records	Pass
3	Provide Valid Password	Password: 12345678			
4	Click on Login button				
5	Provide Invalid UserName or password	User Name: frank@gmail.com Password: 1234	User should not beable to Login	Message for enter validuser or password displayed	Pass
6	Provide Null UserName or Password	UserName:null Password: null			
7	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

5.2.5.1 Login Page Testcase

Code

```
package TestCases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplementation.*;
public class Login {
	static WebDriver driver;
	public static void main(String[] args) {
		driver = DriverSetup.getWebDriver("http://localhost/myhospital/hms/user-
login.php");

	driver.findElement(By.name("username")).sendKeys("frankmathewsthomas@gmail.c
om");
		driver.findElement(By.name("password")).sendKeys("12345678");
		driver.findElement(By.name("submit")).click();
	}
}
package browserimplementation;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class DriverSetup {
	public static String projectPath = System.getProperty("user.dir");
	public static WebDriver getWebDriver(String baseUrl) {
		System.setProperty("webdriver.chrome.driver",
projectPath+"\\drivers\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
		driver.get(baseUrl);
		return driver;
	}
}
```

Output:



5.2.5.2 Update Profile Testcase

Test Case 2					
Project Name: Hospital Management System					
Update Profile Test Case					
Test Case ID: updateProfile			**Test Designed By: Frank Mathews Thomas**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 24-05-2022		
Module Name:Login Screen					
Test Title : Update profile details			**Test Execution Date:** 24-05-2022		
Description: Login to system and update Profile information, if some error occurs, test will fail					
Pre-Condition : User has valid user name and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to LoginPage		Login Page shouldbe	Login page displayed	Pass
2	Provide Valid Username	User Name: frank@gmail.com			

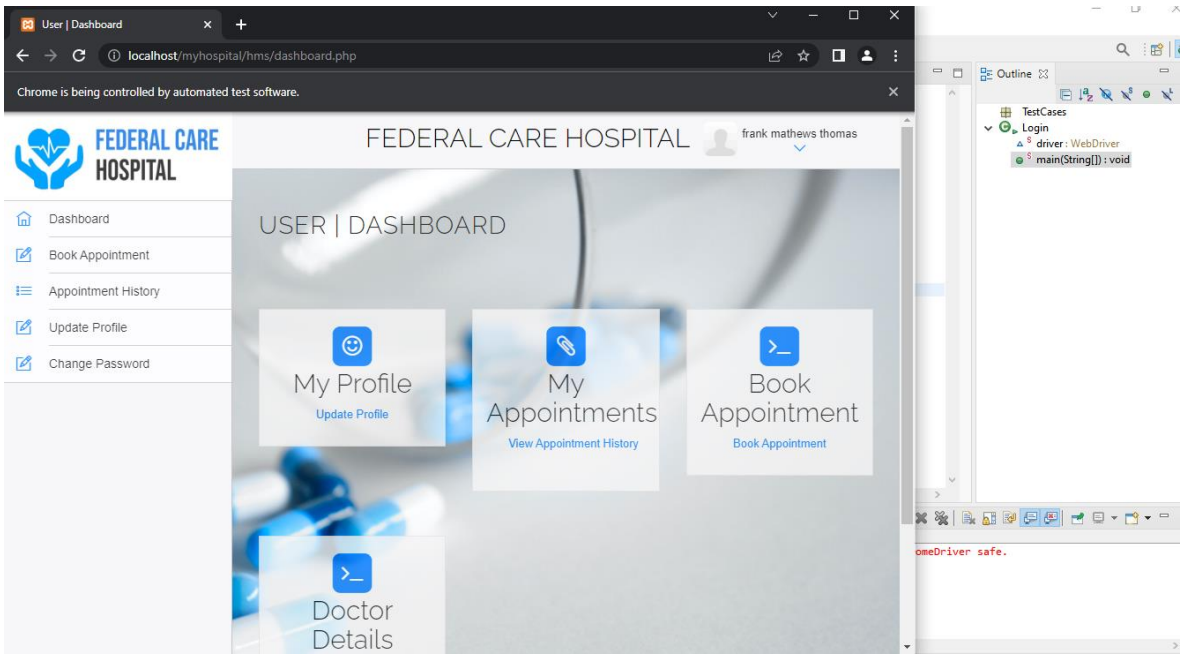
3	Provide Valid Password	Password: 12345678	User should be ableto Login	User Logged in and navigated to Patient Dashboard withrecords	Pass
4	Click on Login button				
5	Provide Profile informations	Input Profile details	User will be redirected to Patient dashboard	User will be redirected to Patient dashboard	Pass
7	Click on Update button				
8	Provide invalid informations	Input invalid Profile details	User will be redirected to Patient dashboard	User will be stay on that page showing error message	Pass
9	Click on Update button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;

public class UpdateProfile {
public static WebDriver driver;
public static void main(String[] args) {
driver =
DriverSetup.getWebDriver("http://localhost/myhospital/login.php");
driver.findElement(By.name("login")).click();

driver.findElement(By.name("userName")).sendKeys("frank@gmail.com"
);
driver.findElement(By.name("password")).sendKeys("12345678");
driver.findElement(By.name("submitButton")).click();

driver.get("http://localhost/myhospital/hms/updateprofile.php");
```

```
driver.findElement(By.name("fname")).sendKeys("Frank");
driver.findElement(By.name("address")).sendKeys("Paramthottu");
driver.findElement(By.name("city")).sendKeys("Pala");
driver.findElement(By.name("dob")).sendKeys("17/09/98");
driver.findElement(By.name("gender")).sendKeys("Male");
driver.findElement(By.name("bloodgrp")).sendKeys("B-ve");
driver.findElement(By.name("submit")).click();


String actualUrl="http://localhost/myhospital/hms/dashboard.php";
String expectedUrl= driver.getCurrentUrl();

if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
} else {
System.out.println("Test failed");
}
}
}
```

Output:



CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development

project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to book and by viewing profile of Doctors. Provides lots of advantages like search doctors, view profile of doctors, enhanced user interface, payment options, add feedback, daily progress report option, complaint status and may more. Implementation of hospital management system project helps to store all the kinds of records, provide coordination and user communication, implement policies, improve day-to-day operations, arrange the supply chain, manage financial and human resources, and market hospital services .Hospital management system is all about the modernizing a hospital through use of technology. Computers helps in it and take over the manual system for quick and easy functioning. This hospital management system is a quite the reliable and is proven on many stages. All the basic requirements of the hospital are provided in the hospital in order to manage it perfectly and large amount of data can also be stored . It gives many facilities like searching for the detail of patient as well as the creation of test reports. So it is an important system for modern days.

.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode.
- Customers can able to do advanced search options
- Customers can able to add complaints and feedbacks etc.
- Doctors can able to view schedule details and add daily progress report etc.
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES

- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Roger S Pressman, “Software Engineering”, 1994.
- Pankaj Jalote, “Software engineering: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES

- www.w3schools.com
- www.jquery.com
- https://www.lovelycoding.org/construction-work-management-system/
- https://www.slideshare.net/chiragbarasiya/construction-management-systemfinal-yearreport
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-8301998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 SAMPLE CODE

Home

```
<!DOCTYPE HTML>
<html>
<head>
<title>FEDERAL CARE  Hospital</title>
<link href="css/style.css" rel="stylesheet" type="text/css"  media="all" />
</head>

<body  class="bgimg" bgcolor="#8e8e93">
        <div class="header">
<div class="wrap">
<div class="logo">
<img src="images\logo1.png" style="height:100px;width:300px"/>

</div>
<div class="top-nav">
<ul><li><a href="index.html">Home</a></li>

        <li><a href="contact.php">contact</a></li>

        </ul>
</div>
<div class="clear"> </div>
</div>
</div>
<div class="clear"> </div>
<br><br><br>
<div style="margin-left:10%; margin-right:10%; margin-top:5%;">
<center>
<div class="border" style="display:inline-block;">
<h2 style="font-family: Raleway, sans-serif; font-weight: 300;margin-top: 0;color:
#5b5b60;"><font size="7px;"><center>Patient Login</font><h2>
```

```
<img src="images\grid-img3.png">
<br>
<font size="4px;">
Login to Book Appointment
</font>
<br><br><br>
<a href="hms/user-login.php">
<button class="btn" type="button" name="login">
Login
</button></a>
<a href="hms/registration.php">
<button class="btn" type="button" name="register">
Register
</button></a>
</center>
</div>
<div class="border" style="display:inline-block; margin-left:15%;">
        <h2 style="font-family: Raleway, sans-serif; font-weight:
        300;margin-top: 0;color:
#5b5b60;"><font size="10px;"><center>Doctor Login</font><h2>
<img src="images\grid-img1.png">
<br>
<font size="4px;">
Login to View Appointments
</font>
<br>
<a href="hms/doctor/index.php">
<br><br>
<button class="btn" type="button" name="login">
Login
</button></a>
</center>
</div>
</center>
```

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

Patient Login

```
<?php
include_once('include/config.php'); if(isset($_POST['submit']))
{
$fname=$_POST['full_name'];
$address=$_POST['address'];
$city=$_POST['city'];
$dob=$_POST['dob'];
$gender=$_POST['gender'];
$bloodgrp=$_POST['blg'];
$email=$_POST['email'];
$password=md5($_POST['password']);
$query=mysqli_query($con,"insertinto
users(fullname,address,city,dob,gender,bloodgrp,email,password)
values('$fname','$address','$city','$dob','$gender','$bloodgrp','$email','$password')");
if($query) {
        //echo "<script>alert('Successfully Registered. You can login now');</script>";
        header('location:./user-login.php');

}
}
?>
```

ADMIN Login

```
<?php
session_start(); error_reporting(0); include("include/config.php"); if(isset($_POST['submit']))
{
$ret=mysqli_query($con,"SELECT*FROMadminWHERE
        username='".$_POST['username']."' and password='".$_POST['password']."'");
$num=mysqli_fetch_array($ret); if($num>0)
```

```
{
$extra="dashboard.php";//
$_SESSION['login']=$_POST['username'];
$_SESSION['id']=$num['id'];
$host=$_SERVER['HTTP_HOST'];
$uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
header("location:http://$host$uri/$extra"); exit();
}
else
{
$_SESSION['errmsg']="Invalid username or password";
$extra="index.php";
$host  = $_SERVER['HTTP_HOST'];
$uri                    =                    rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
header("location:http://$host$uri/$extra");
exit(); }
}
?>
```

Dashboard -Patient

```
<?php
session_start();
include('include/config.php');
include('include/checklogin.php');
check_login();

?>
<!DOCTYPE html>
<html lang="en">
<head>
<title>User  | Dashboard</title>
<meta charset="utf-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimum-scale=1.0, maximum-scale=1.0">
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta content="" name="description" />
<meta content="" name="author" />
<link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|Raleway:300,400,500,600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
<link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
<link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
<link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
<link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
<link href="vendor/bootstrap-touchspin/jquery.bootstrap-touchspin.min.css" rel="stylesheet" media="screen">
<link href="vendor/select2/select2.min.css" rel="stylesheet" media="screen">
<link href="vendor/bootstrap-datepicker/bootstrap-datepicker3.standalone.min.css" rel="stylesheet" media="screen">
<link href="vendor/bootstrap-timepicker/bootstrap-timepicker.min.css" rel="stylesheet" media="screen">
<link rel="stylesheet" href="assets/css/styles.css">
<link rel="stylesheet" href="assets/css/plugins.css">
<link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />

</head>
<body>
<div id="app">
<?php include('include/sidebar.php');?>
<div class="app-content">

<?php include('include/header.php');?>
```

```
<div class="main-content" >
<div class="wrap-content container" id="container">

<section id="page-title">
<div class="row">
<div class="col-sm-8">
<h1 class="mainTitle">User | Dashboard</h1>

        </div>
<ol class="breadcrumb">
<li>
<span></span>
</li>

</ol>
</div>
</section>

<div class="container-fluid container-fullw bg-white">
<div class="row">
<div class="col-sm-4">
<div class="panel panel-white no-radius text-center">
<div class="panel-body">
<span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-smile-o fa-stack-1x fa-inverse"></i> </span>
<h2 class="StepTitle">My Profile</h2>

        <p class="links cl-effect-1">
        <a href="edit-profile.ph
    Update Profile
</a>
</p>
```

```
</div>
        </div>
</div>
<div class="col-sm-4">
<div class="panel panel-white no-radius text-center">
<div class="panel-body">
<span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-paperclip fa-stack-1x fa-inverse"></i> </span>
<h2 class="StepTitle">My Appointments</h2>

<p class="cl-effect-1">
        <a href="appointment-history.php">
                View Appointment History
</a>
</p>
</div>
</div>
</div>
<div class="col-sm-4">
<div class="panel panel-white no-radius text-center">
div class="panel-body">
<span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-terminal fa-stack-1x fa-inverse"></i> </span>

<h2 class="StepTitle"> Book Appointment</h2>
<p class="links cl-effect-1">
<a href="book-appointment.php">
Book Appointment

</a>
</p>

</div>
</div>
```

```
</div>
</div>
</div>
<div class="col-sm-4">
<div class="panel panel-white no-radius text-center">
<div class="panel-body">
<span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-terminal fa-stack-1x fa-inverse"></i> </span>
<h2 class="StepTitle"> Doctor Details</h2>

<p class="links cl-effect-1">
<a href="doctor-details.php">Doctor Details</a>
</p>
</div>
</div>
</div>
</div>
</div>
        </div>
        </div>
        </div>
        <?php include('include/footer.php');?>
        <?php include('include/setting.php');?>

                </div>
                <script src="vendor/jquery/jquery.min.js"></script>
                <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
                <script src="vendor/modernizr/modernizr.js"></script>
                <script src="vendor/jquery-cookie/jquery.cookie.js"></script>
                <script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
                <script src="vendor/switchery/switchery.min.js"></script>

                <script src="vendor/maskedinput/jquery.maskedinput.min.js"></script>
```

```
                <script src="vendor/bootstrap-touchspin/jquery.bootstrap-
touchspin.min.js"></script>
                <script src="vendor/autosize/autosize.min.js"></script>
                <script src="vendor/selectFx/classie.js"></script>
                <script src="vendor/selectFx/selectFx.js"></script>
                <script src="vendor/select2/select2.min.js"></script>
                <script src="vendor/bootstrap-datepicker/bootstrap-
datepicker.min.js"></script>
                <script src="vendor/bootstrap-timepicker/bootstrap-
timepicker.min.js"></script>

                <script src="assets/js/main.js"></script>

                <script src="assets/js/form-elements.js"></script>
                <script>
                        jQuery(document).ready(function() {
                                Main.init();
                                FormElements.init();
                        });
                </script>

        </body>
</html>
```

Dashboard -Doctor

```
 <?php
session_start();

include('include/config.php');
include('include/checklogin.php');
check_login();

?>
```

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Doctor  | Dashboard</title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimum-scale=1.0, maximum-scale=1.0">
        <meta name="apple-mobile-web-app-capable" content="yes">
        <meta name="apple-mobile-web-app-status-bar-style" content="black">
        <meta content="" name="description" />
        <meta content="" name="author" />
        <link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|Raleway:300,400,500,600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
        <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
        <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
        <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
        <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
        <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
        <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-touchspin/jquery.bootstrap-touchspin.min.css" rel="stylesheet" media="screen">
        <link href="vendor/select2/select2.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-datepicker/bootstrap-datepicker3.standalone.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-timepicker/bootstrap-timepicker.min.css" rel="stylesheet" media="screen">
        <link rel="stylesheet" href="assets/css/styles.css">
        <link rel="stylesheet" href="assets/css/plugins.css">
        <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />
```

```
        </head>
        <body>
                <div id="app">
<?php include('include/sidebar.php');?>
<div class="app-content">

<?php include('include/header.php');?>



<div class="main-content" >
<div class="wrap-content container" id="container">

<section id="page-title">
<div class="row">
<div class="col-sm-8">
<h1 class="mainTitle">Doctor | Dashboard</h1>
</div>
<ol class="breadcrumb">
<li>

</ol>
</div>
</section>

<div class="container-fluid container-fullw bg-white">
<div class="row">
<div class="col-sm-4">
<div class="panel panel-white no-radius text-center">
<div class="panel-body">
<span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-smile-o fa-stack-1x fa-inverse"></i> </span>
<h2 class="StepTitle">My Profile</h2>
```

```
<p class="links cl-effect-1">
<a href="edit-profile.php">

        Update Profile
</a>
</p>
</div>
                        </div>
                        </div>
                <div class="col-sm-4">
        <div class="panel panel-white no-radius text-center">
        <div class="panel-body">
<span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-paperclip fa-stack-1x fa-inverse"></i> </span>
<h2 class="StepTitle">My Appointments</h2>

<p class="cl-effect-1">
<a href="appointment-history.php">

        View Appointment History
</a>
</p>
</div>
</div>
</div>

</div>
</div>
</div>
</div>
</div>

        <?php include('include/footer.php');?>
```

```
<?php include('include/setting.php');?>
                    <>

            </div>

            <script src="vendor/jquery/jquery.min.js"></script>
            <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
            <script src="vendor/modernizr/modernizr.js"></script>
            <script src="vendor/jquery-cookie/jquery.cookie.js"></script>
            <script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
            <script src="vendor/switchery/switchery.min.js"></script>

            <script src="vendor/maskedinput/jquery.maskedinput.min.js"></script>
            <script                    src="vendor/bootstrap-touchspin/jquery.bootstrap-
touchspin.min.js"></script>
            <script src="vendor/autosize/autosize.min.js"></script>
            <script src="vendor/selectFx/classie.js"></script>
            <script src="vendor/selectFx/selectFx.js"></script>
            <script src="vendor/select2/select2.min.js"></script>
            <script                      src="vendor/bootstrap-datepicker/bootstrap-
datepicker.min.js"></script>
            <script                     src="vendor/bootstrap-timepicker/bootstrap-
timepicker.min.js"></script>

            <script src="assets/js/main.js"></script>

            <script src="assets/js/form-elements.js"></script>
            <script>
                jQuery(document).ready(function() {
                    Main.init();
                    FormElements.init();
                });
            </script>
```

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

Video Consultation

Index.html

```
<!DOCTYPE html>
<head>
   <title>Federal Care Hospital</title>
   <meta charset="utf-8" />
   <link type="text/css" rel="stylesheet" href="https://source.zoom.us/2.3.5/css/bootstrap.css"
/>
   <link type="text/css" rel="stylesheet" href="https://source.zoom.us/2.3.5/css/react-
select.css" />
   <meta name="format-detection" content="telephone=no">
   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">

</head>

<body>
   <style>
      .sdk-select {
         height: 34px;
         border-radius: 4px;
      }

      .websdktest button {
         float: right;
         margin-left: 5px;
      }

      #nav-tool {
         margin-bottom: 0px;
      }
```

```
    #show-test-tool {
        position: absolute;
        top: 100px;
        left: 0;
        display: block;
        z-index: 99999;
    }

    #display_name {
        width: 250px;
    }

    #websdk-iframe {
        width: 700px;
        height: 500px;
        border: 1px;
        border-color: red;
        border-style: dashed;
        position: fixed;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
        left: 50%;
        margin: 0;
    }
</style>

<nav id="nav-tool" class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">Federal Care Hospital</a>
        </div>
        <div id="navbar" class="websdktest">
            <form class="navbar-form navbar-right" id="meeting_form">
```

```
<div class="form-group">
  <input type="text" name="display_name" id="display_name" value="2.3.5#CDN" maxLength="100"
    placeholder="Name" class="form-control" required>
</div>
<div class="form-group">
  <input type="text" name="meeting_number" id="meeting_number" value="" maxLength="200"
    style="width:150px" placeholder="Meeting Number" class="form-control" required>
</div>
<div class="form-group">
  <input type="text" name="meeting_pwd" id="meeting_pwd" value="" style="width:150px"
    maxLength="32" placeholder="Meeting Password" class="form-control">
</div>
<div class="form-group">
  <input type="text" name="meeting_email" id="meeting_email" value="" style="width:150px"
    maxLength="32" placeholder="Email option" class="form-control">
</div>

<div class="form-group">
  <select id="meeting_role" class="sdk-select">
    <option value=0>Attendee</option>
    <option value=1>Host</option>
    <option value=5>Assistant</option>
  </select>
</div>
<div class="form-group">
  <select id="meeting_china" class="sdk-select">
    <option value=0>Global</option>

  </select>
```

```
            </div>
            <div class="form-group">
              <select id="meeting_lang" class="sdk-select">
                <option value="en-US">English</option>
                <option value="de-DE">German Deutsch</option>
                <option value="es-ES">Spanish Español</option>
                <option value="fr-FR">French Français</option>
                <option value="jp-JP">Japanese 日本語</option>
                <option value="pt-PT">Portuguese Portuguese</option>
                <option value="ru-RU">Russian Русский</option>
                <option value="zh-CN">Chinese 简体中文</option>
                <option value="zh-TW">Chinese 繁体中文</option>
                <option value="ko-KO">Korean 한국어</option>
                <option value="vi-VN">Vietnamese Tiếng Việt</option>
                <option value="it-IT">Italian italiano</option>
              </select>
            </div>

            <input type="hidden" value="" id="copy_link_value" />
            <a href="./meeting.html" class="btn btn-success">Join</a>
            <button type="submit" class="btn btn-primary" id="clear_all">Clear</button>
            <button type="button" link="" onclick="window.copyJoinLink('meeting.html')"
              class="btn btn-primary" id="copy_join_link">Copy Direct join link</button>
          </form>
        </div>
        <!--/.navbar-collapse -->
      </div>
    </nav>
    <div id="show-test-tool">
      <button type="submit" class="btn btn-primary" id="show-test-tool-btn"
        title="show or hide top test tool">Show</button>
    </div>
```

```
    <script>
        document.getElementById('show-test-tool-btn').addEventListener("click", function (e) {
            var textContent = e.target.textContent;
            if (textContent === 'Show') {
                document.getElementById('nav-tool').style.display = 'block';
                document.getElementById('show-test-tool-btn').textContent = 'Hide';
            } else {
                document.getElementById('nav-tool').style.display = 'none';
                document.getElementById('show-test-tool-btn').textContent = 'Show';
            }
        })
    </script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/react.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/react-dom.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/redux.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/redux-thunk.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/lodash.min.js"></script>
    <script src="https://source.zoom.us/zoom-meeting-2.3.5.min.js"></script>
    <script src="js/tool.js"></script>
    <script src="js/vconsole.min.js"></script>
    <script src="js/index.js"></script>
    <script>
    </script>
</body>
</html>
```

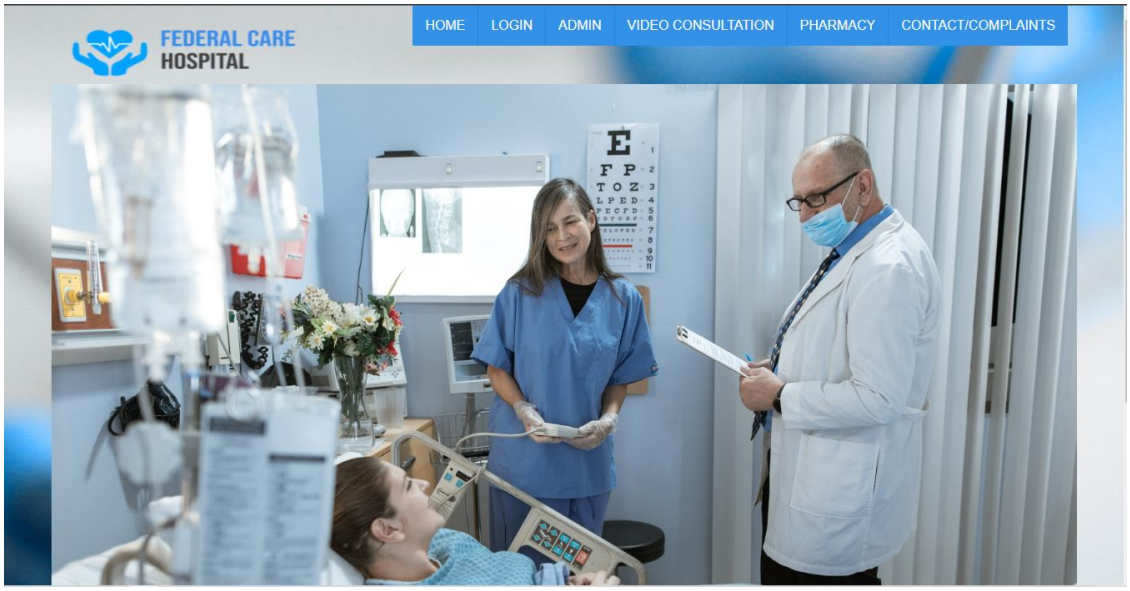
Meeting.html

```
<!DOCTYPE html>
<head>
    <title>Federal Care Hospital</title>
    <meta charset="utf-8" />
    <link type="text/css" rel="stylesheet" href="https://source.zoom.us/2.3.5/css/bootstrap.css"
/>
```

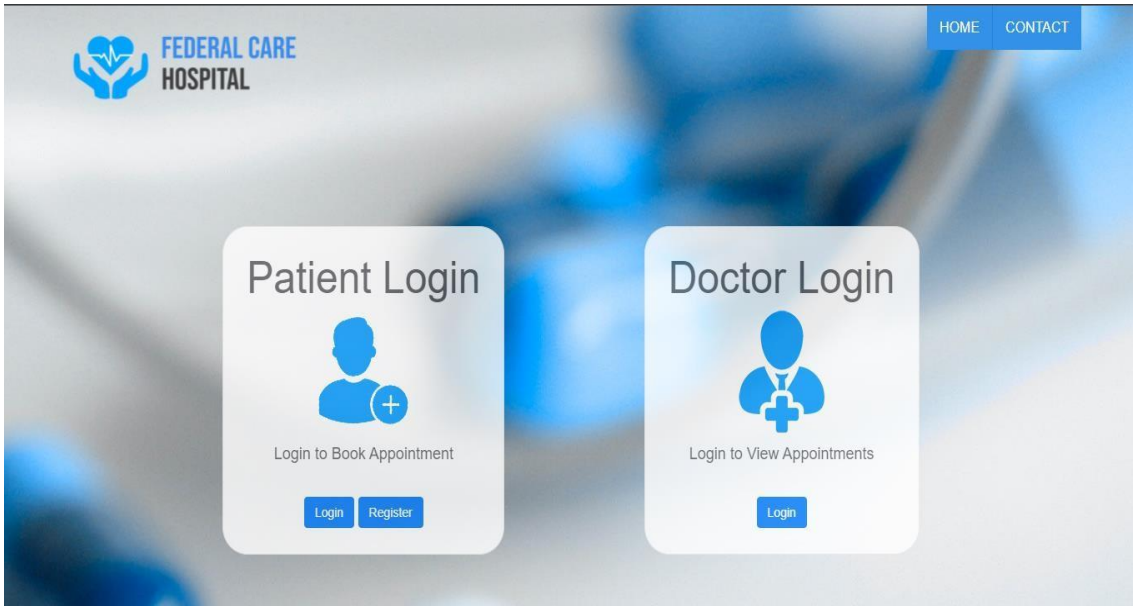
```
    <link type="text/css" rel="stylesheet" href="https://source.zoom.us/2.3.5/css/react-select.css" />
    <meta name="format-detection" content="telephone=no">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <meta http-equiv="origin-trial" content="">
</head>

<body>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/react.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/react-dom.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/redux.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/redux-thunk.min.js"></script>
    <script src="https://source.zoom.us/2.3.5/lib/vendor/lodash.min.js"></script>
    <script src="https://source.zoom.us/zoom-meeting-2.3.5.min.js"></script>
    <script src="js/tool.js"></script>
    <script src="js/vconsole.min.js"></script>
    <script src="js/meeting.js"></script>
    <script>
    </script>
</body>
</html>
```

9.2 SCREENSHOTS

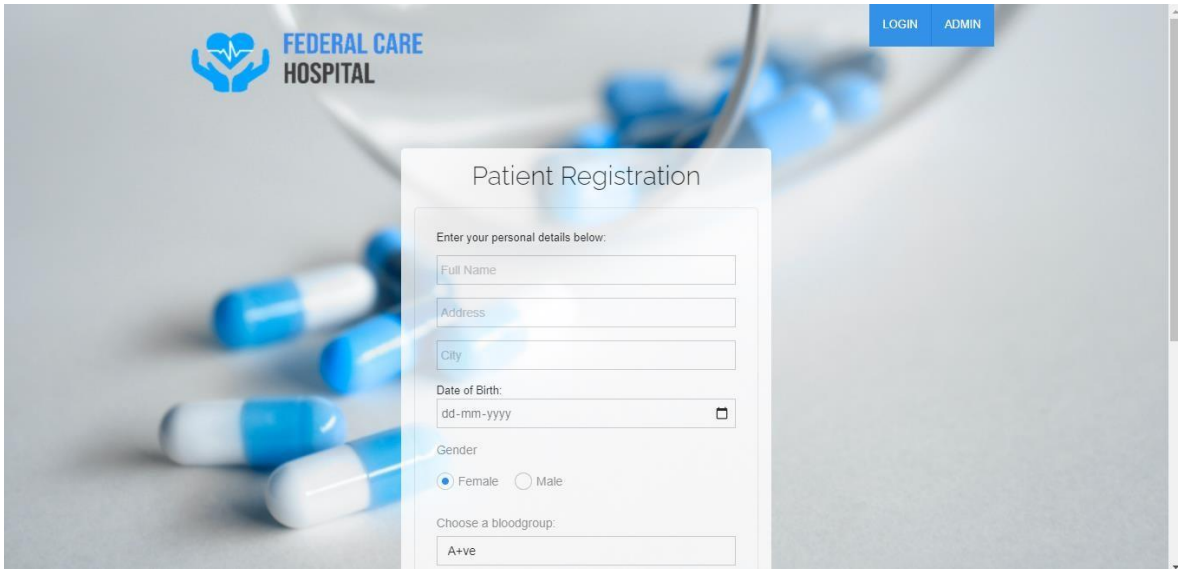
Index page



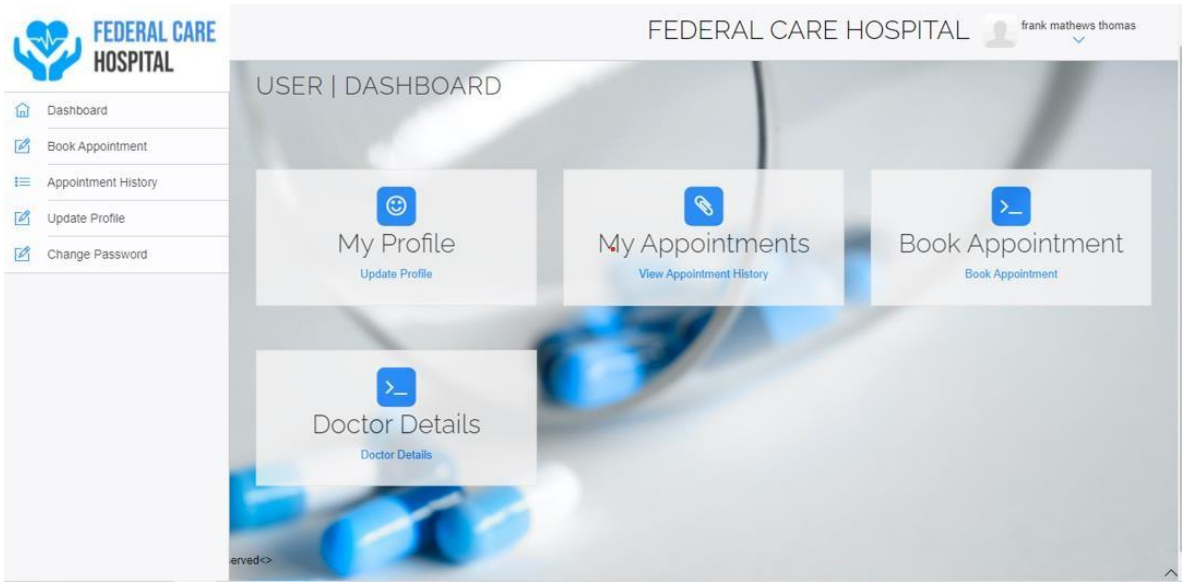
Login page



Patient registration

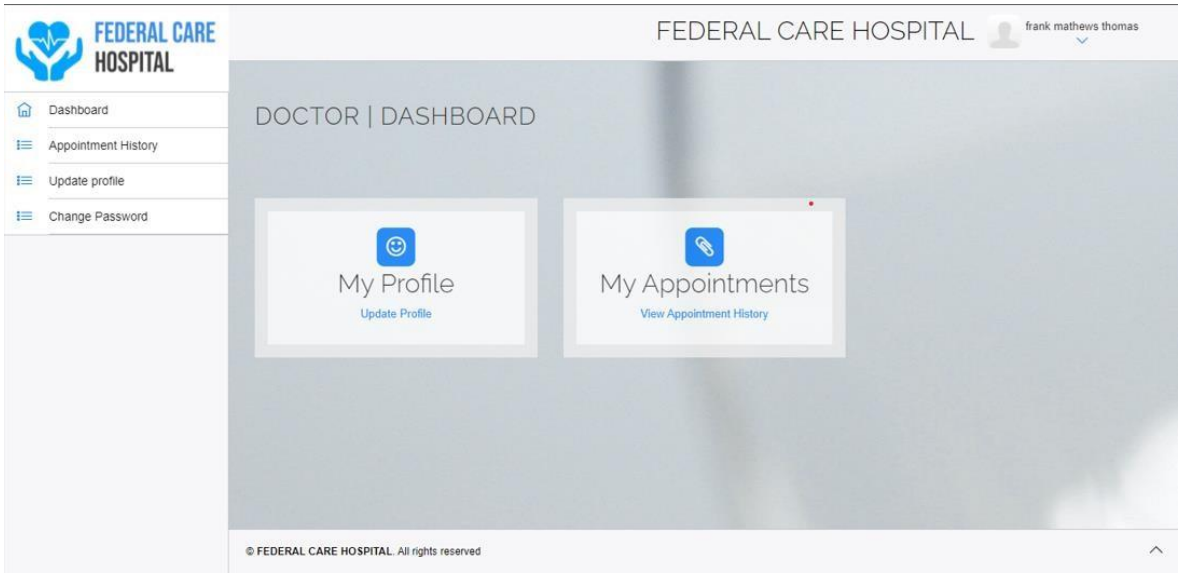


Patient dashboard

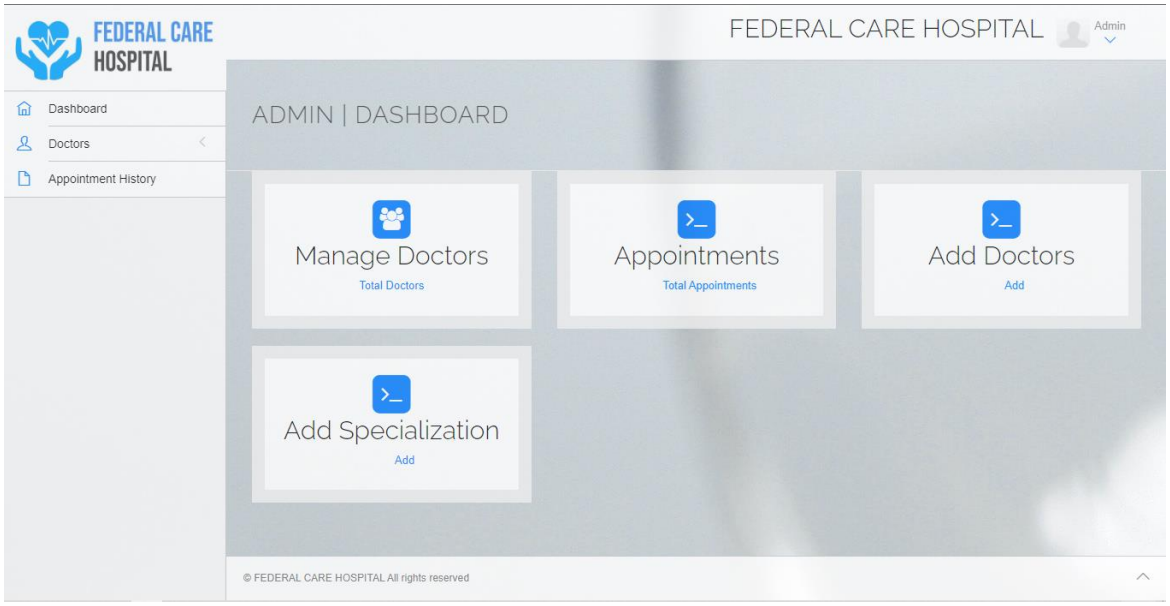


Patient appointment history

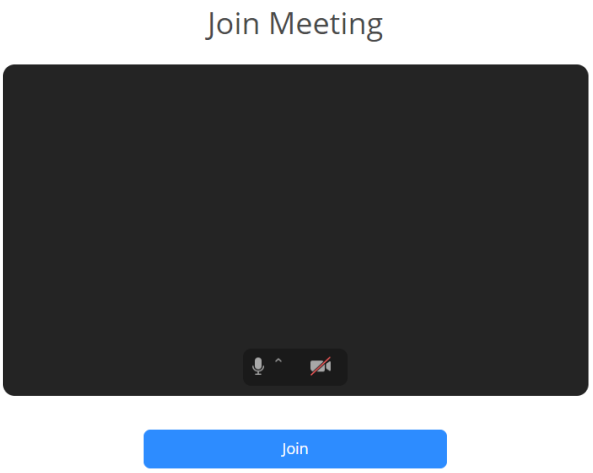
Doctor dashboard



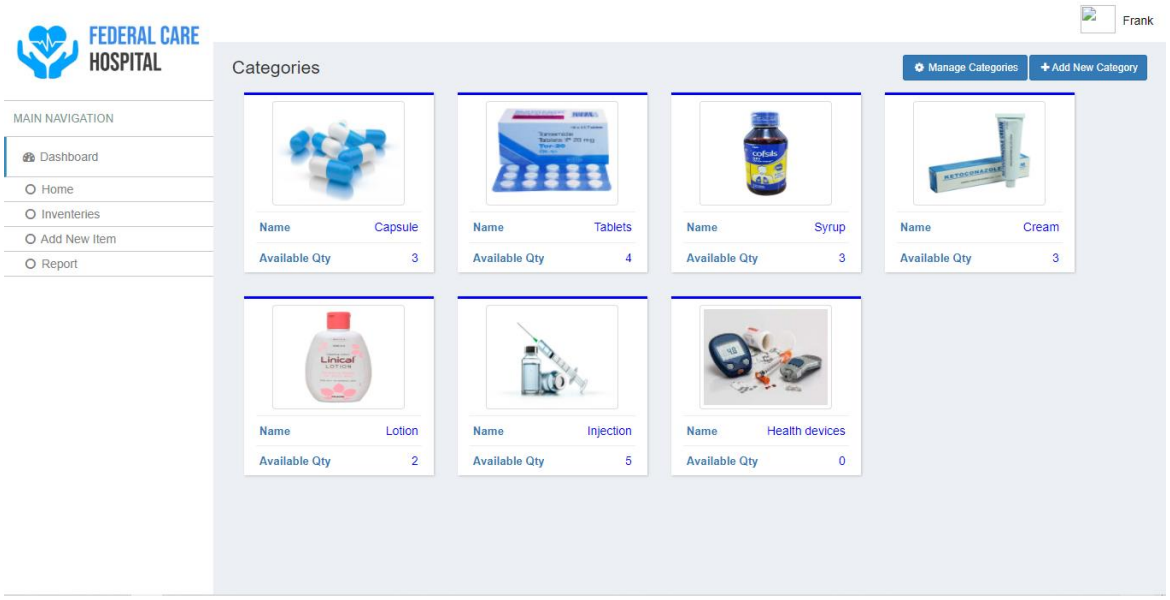
Admin dashboard



Video consultation


Join Meeting
Join

Pharmacy


FEDERAL CARE HOSPITAL
MAIN NAVIGATION
Dashboard
Home
Inventeries
Add New Item
Report
Frank
Categories
Manage Categories
+ Add New Category
Name Capsule
Available Qty 3
Name Tablets
Available Qty 4
Name Syrup
Available Qty 3
Name Cream
Available Qty 3
Name Lotion
Available Qty 2
Name Injection
Available Qty 5
Name Health devices
Available Qty 0