E-HEALTHCARE SYSTEM

A

Project Report

Submitted In Partial Fulfillment of the Requirements

for the award of

Industrial Training Certificate

In

Computer Science and Engineering

Under Guidance Of

Mr. RAHUL SHARMA

MICROSOFT & HPE CERTIFIED TECHNICAL TRAINER

Project Carried Out

At

Academy of Skill Development

(An ISO 9001:2015 Certified)
SDF Building, Module No:103, Ground Floor

Sector V, GP Block, Kolkata- 700091

Submitted By

M GOWRI SHANKAR

ASSISTANT PROFESSOR,
DEPARMENT OF COMPUTER SCIENCE AND
ENGINEERING,
GOVERNMENT COLLEGE OF TECHNOLOGY,
COIMBATORE, TAMILNADU - 641013

1. Title of the Project   : **E-HealthCare Management System**

2. Name of the Guide   : **Mr. Rahul Sharma**

	Ph.D(Reg)	M.Tech.	MCA	M.Sc.	MBA	B-Tech
3. Educational Qualification of the Guide(s)			Y			

4. Working / Teaching experience of the Guide: **Mr. Rahul Sharma (8 Years)**

5. Software used in the Project

- Mongo Db
- Express JS
- ReactJS
- Node Js
- BootStrap

Signature of the Guide

For Office Use Only

Approved		Not Approved

…………………………………………….

Signature, Designation Stamp of the Project Proposal Evaluator

Date: …………………….

Self Certificate

This is to certify that the dissertation/project proposal entitled **"E-HealthCare Management System"** is done by **M GOWRI SHANKAR** an authentic work carried out for the partial fulfillment of the requirements for the award of the **Industrial Training Certificate in Computer Science and Engineering** under the guidance of **Rahul Sharma**. The matter embodied in this project work has not been submitted earlier forward of any degree to the best of my knowledge and belief.

Name of the Student(s):

   1. STUDENT NAME   : M GOWRI SHANKAR

Certificate By Guide

This is to certify that this project entitled **"E-HealthCare Management System"** submitted in partial fulfillment of the INDUSTRIAL TRAINING MANDATE BY AICTE **in Computer Science and Engineering** by **M GOWRI SHANKAR, ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, GOVERNMENT COLLEGE OF TECHNOLOGY, COIMBATORE, TAMILNADU - 641013,** is an authentic work carried out under my guidance & best of our knowledge and belief.

Signature of the student   Signature of the Guide

Date: Date:

Certificate of Approval

This is to certify that this documentation of **Industrial Training Program 2025**, entitled **"E-HealthCare Management System"** is a record of bona-fide work, carried out by **M GOWRI SHANKAR** under my supervision and guidance.

In my opinion, the report in its present form is in fulfillment of all the requirements, as specified by the **AICTE** and as per regulations of the **Academy of Skill Development** In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for **Industrial Training Certificate.**

Mr. Rahul Sharma

Microsoft & HP Certified Technical Trainer At Academy of Skill Development (An ISO 9001:2015 Certified) (Approved by NCVT & Ministry of HRD, Govt of India)

ACKNOWLEDGEMENT

Table of Contents

1. COMPANY PROFILE

Academy Of Skill Development (ASD) - A unit of Ardent Computech Pvt. Ltd., an MSME, ISO 9001:2015, is a non-profit Govt. registered organization and a subsidiary of Ardent Computech Pvt. Ltd. Ardent has MoU with the AICTE NEAT Cell to conduct Industrial Training and Internships for the students and faculties. ASD has affiliations to conduct global certification as an authorized Certiport test center. Mission of ASD is to develop skills of the candidates to meet the dynamics of demand in the job market. We look forward to bridge the gap between the industry and academia. Recent survey has revealed the huge gap between demand and supply of candidates. Today industry is looking for ready-made candidates with right set of skills. Hence employability enhancement Industrial Training and Internships are very important. At ASD we provide the latest and best Skill Development Industrial Training and Internships.

1. INTRODUCTION

The rapid growth in Information & Communication Technology (ICT), and the power of Internet has strongly impacted the business and service delivery models of today 's global environment. Health Management Systems provide the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. E-healthcare Management Systems are in high demand to handle increasing population needs and also aids the practicing doctors and hospital service and support staff with timely service and precision. There are varied metrics available to assess the performance of services like hospital industry, and the successful implementation and usage of Hospital information system forms a crucial role. This desktop application of E-Health Management system will provide comprehensive, effective and efficient solution for carrying out management of hospitals and clinics fulfilling the needs and requirements of all stakeholders such as doctors, patients and staffs.

2. OBJECTIVE

The E-Healthcare Management System has following goals:

- Authorize doctors and assisting staffs to access the medical history of patient efficiently
- Appointment system for Patients, Easy lookup of prescriptions
- To provide a system that stores patient management data at single place, for a chain of hospitals.
- Each Doctor efficiently manages the Patients appointments.
- To let the doctors and surgeons build their profile and personal repository in which they may store their routine medicines names, precautions instructions and can take required help from this repository later when needed.

2.1 HIGH LEVEL FEATURES TO ACHIEVE GOALS AND OBJECTIVES

1. Patient Registration and Book Appointment
2. Doctor Login and View Appointments
3. Patient Diagnosis.
4. Admin Login – Create / Update Doctor Profiles.

2.2 NON-FUNCTIONAL REQUIREMENTS

1. Availability
2. Secure Storage and Access
3. Reliability
4. Response time
5. Scalability

4. SYSTEM ANALYSIS

4.1. IDENTIFICATION OF NEED

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 studies to the minutest detail and analyzed. 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 organization are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and Decisional variables, analysis 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 area 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.

4.2. FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work

Effort and time spent 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 area that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

a) **Operational Feasibility:** By using this system different types of organization can handle their portal dynamically. There will be one user in this system called admin. To manage portal he can manage doctor, patient can know current status of portal, can see periodically report. Also, can take decision based on Patient and Doctor status: -

- Easy access to the system.

- Doctors of all departments available in the portal.

- Portal is updated regularly.

b) **Technical Feasibility:** To run developed system which is a web-based system we need just one computer and one server. Where PC is in the client end and server is in server end. All the information pushed through the system will stored in server. According necessity data will be fetched from server to the client end.

Hardware	PC/LAPTOP (1 GB RAM and 1TB HARD DISK)
Software	- Mongo dB - Nodejs - VS Code

b) **Economic feasibility:** As we want to build this project for managing patient we just need one pc and one server. So, we can manage it in low cost because we don't need a large number of pc and servers.

4.3. WORK FLOW

The system begins when staff enters information of products after receiving them from supplier. For the first time of key in, the system will consider it as process of getting the data into database. Once the code is entered, system will be updated. Then, in module out, the system will reflect as the item is moving out from the database. Usually there are two reasons why the product is out of the system, the product is sold to customer or sent back to supplier because of labefaction. When the product is sold, it will go to process of sale that will calculate the price and amount of profit. However, if the product is out from system because of return to supplier, it will go to manual process.
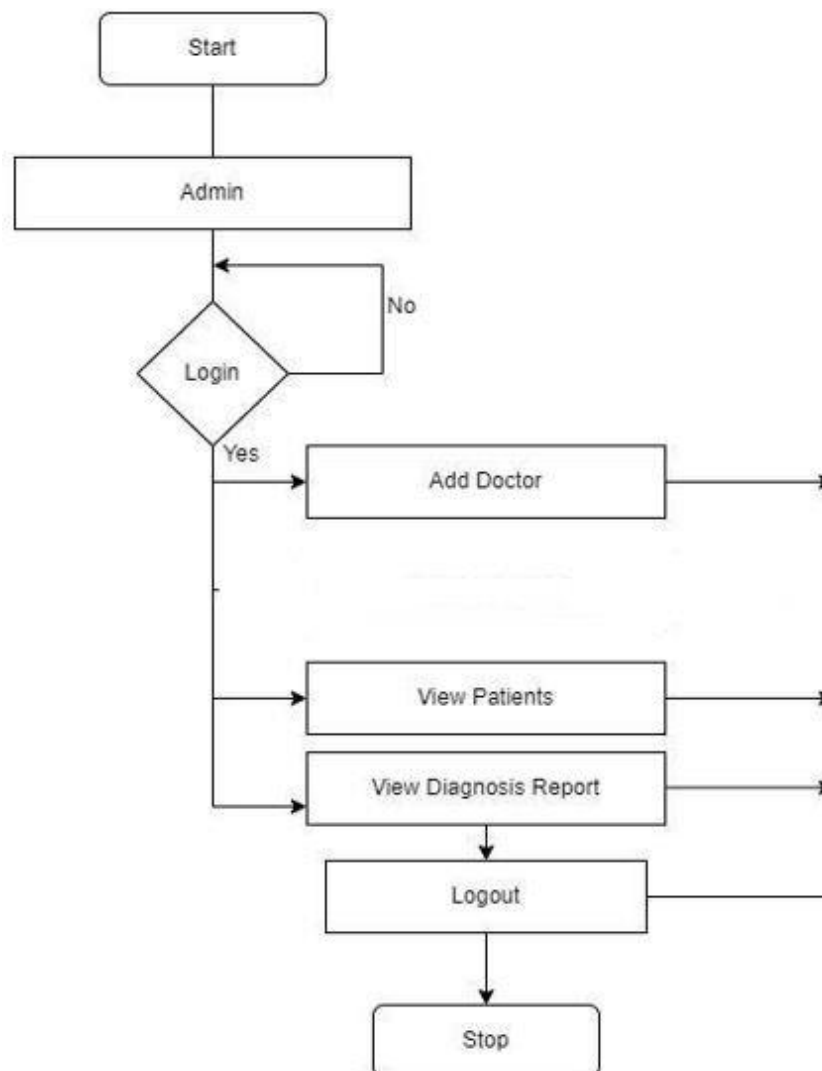
4.4. HARDWARE AND SOFTWARE REQUIREMENTS

4.4.1. Hardware Requirements

- Standard computer with at least i3 processor
- Standard computer with 2GB of RAM
- Standard computer with 100GB of free space
- Active Internet Connectivity with good bandwidth

4.4.2. Software Requirements
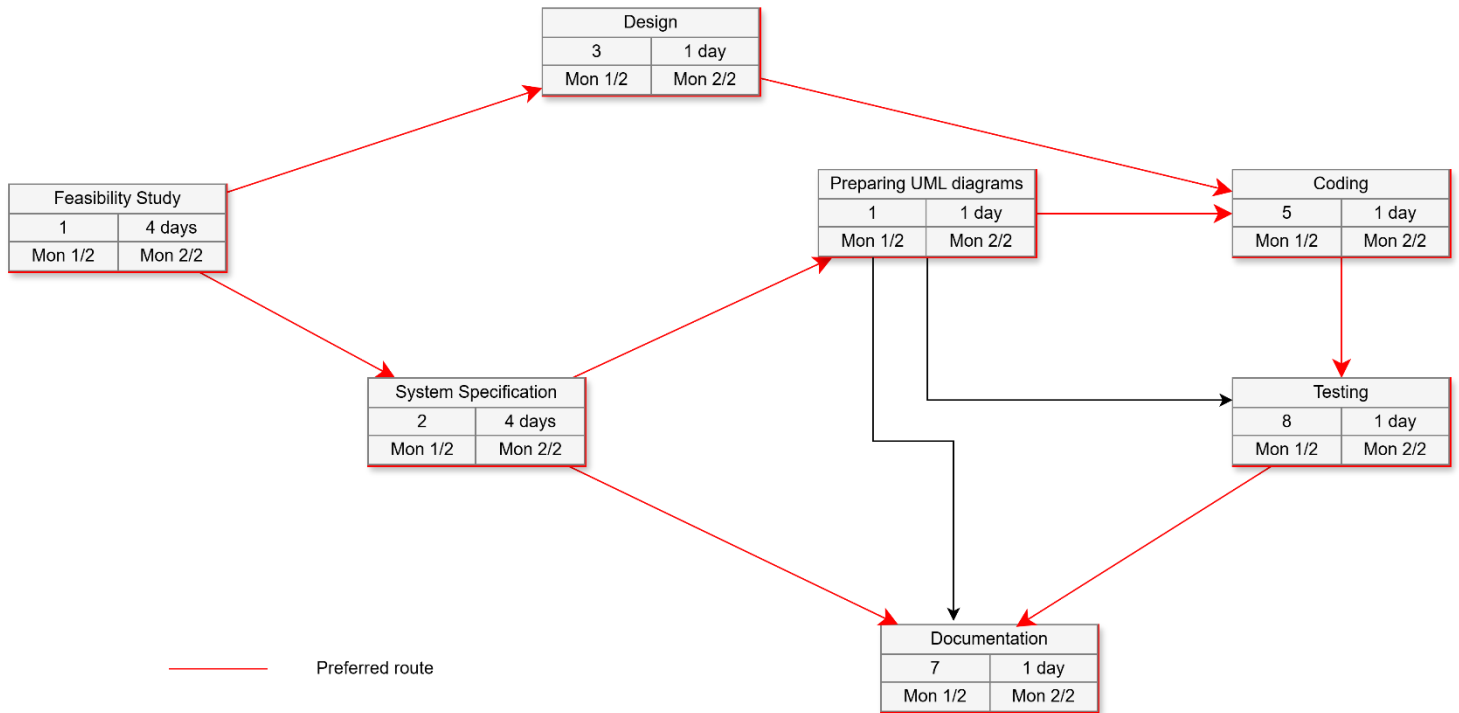
- VS Code
- Mongo dB
- Node js

5. SYSTEM DESIGN

5.1   GANTT CHART

TASK	Start Date	Duration
Feasibility Study	March 1, 2025	4 days
System Specification	March 5, 2025	4 days
Preparing UML diagrams	March 9, 2025	6 days
Design	March 15, 2025	2 days
Coding	March 17, 2025	2 days
Testing	March 19,2025	1 day
Documentation	March 20, 2025	2 days

Gantt Chart Data

Task Name	Mar 1-4	Mar 5-8	Mar 9-14	Mar 15-16	Mar 17-18	Mar 19	Mar 20-21
Feasibility Study							
System Specification							
Preparing UML diagrams							
Design							
Coding							
Testing							
Documentation							

5.2.   PERT CHART

Design	
3	1 day
Mon 1/2	Mon 2/2

Feasibility Study	
1	4 days
Mon 1/2	Mon 2/2

Preparing UML diagrams	
1	1 day
Mon 1/2	Mon 2/2

Coding	
5	1 day
Mon 1/2	Mon 2/2

System Specification	
2	4 days
Mon 1/2	Mon 2/2

Testing	
8	1 day
Mon 1/2	Mon 2/2

————— Preferred route

Documentation	
7	1 day
Mon 1/2	Mon 2/2

Pert Chart

5.3. DATA FLOW DIAGRAM
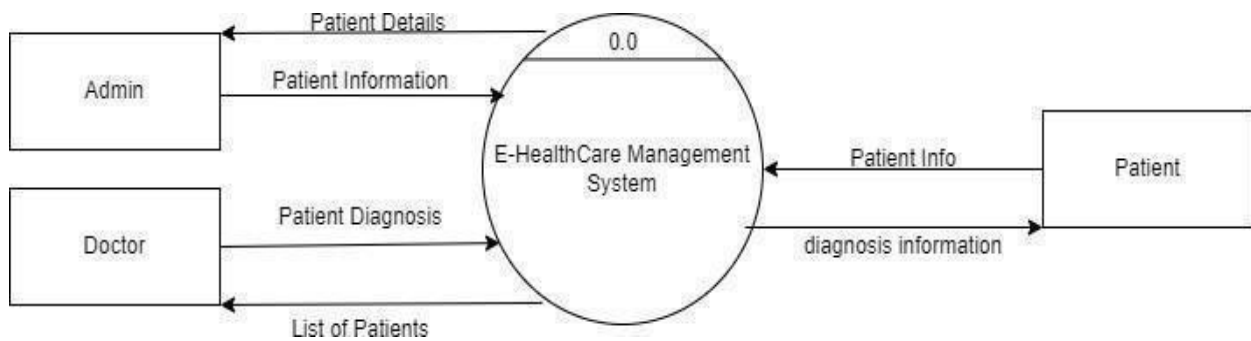
LEVEL 0 DFD
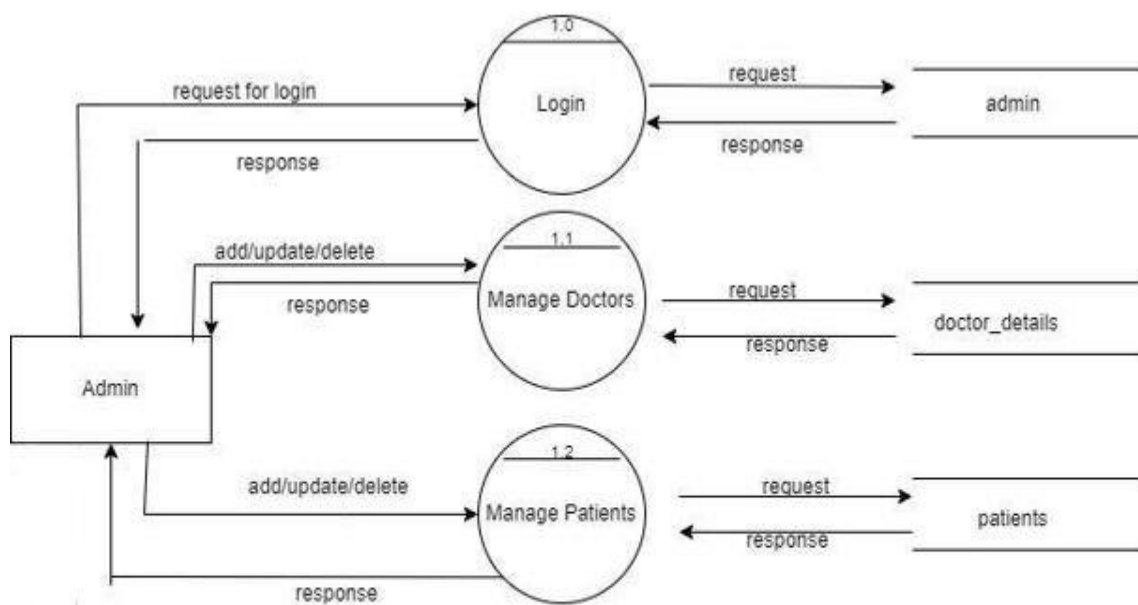
LEVEL 1 DFD ADMIN
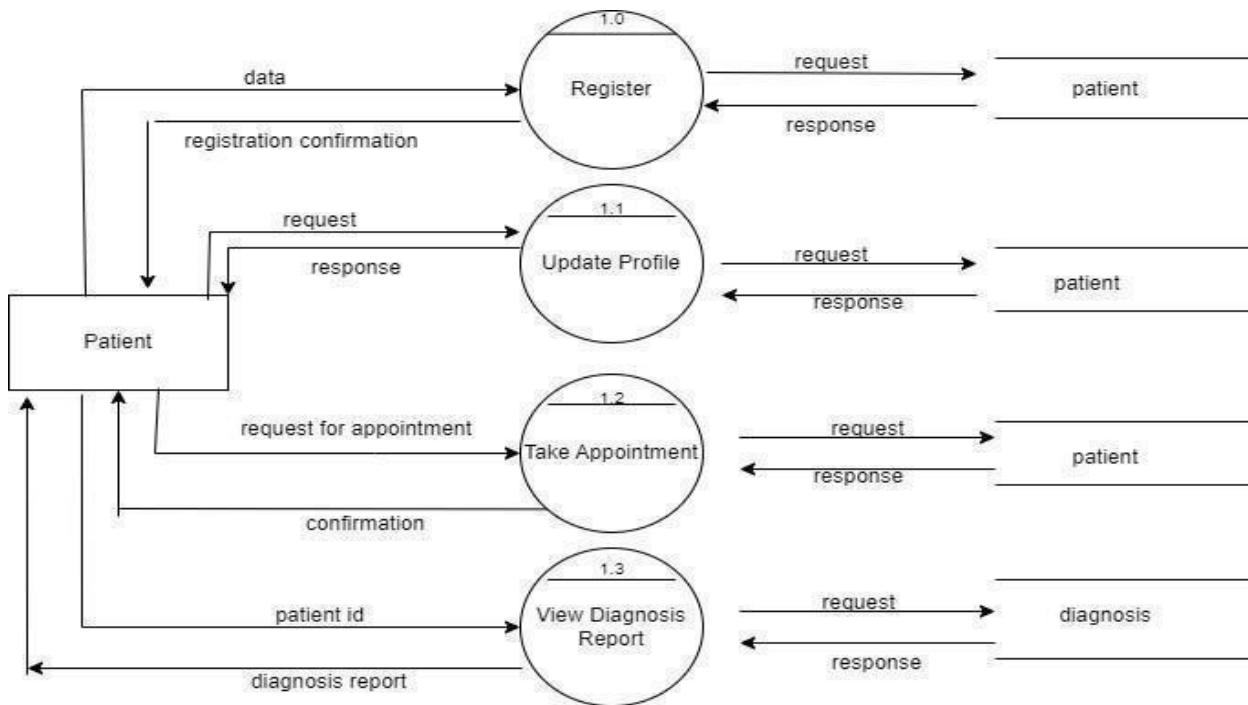
LEVEL 1 DFD FOR PATIENT
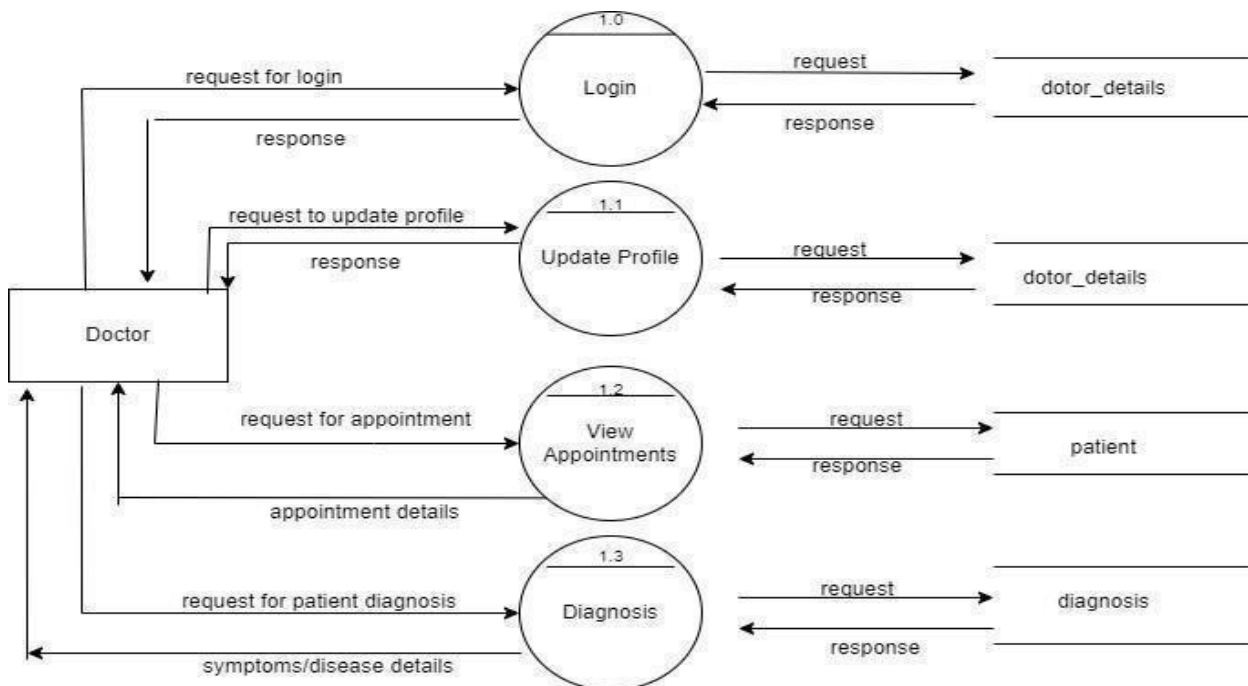
LEVEL 1 DFD FOR DOCTOR

5.4 SEQUENCE DIAGRAM

Sequence diagrams can be useful reference diagrams for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.

- Model the logic of a sophisticated procedure, function, or operation.

- See how tasks are moved between objects or components of a process.

- Plan and understand the detailed functionality of an existing or future scenario.

Popular Sequence Diagram Uses

- **Usage Scenario** – A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system.

- **Method Logic** - Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to Usage Scenario - A usage scenario is a diagram of how your system could potentially be used. It's a great explore the logic of any function, procedure, or complex process.

- **Service Logic** - If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.
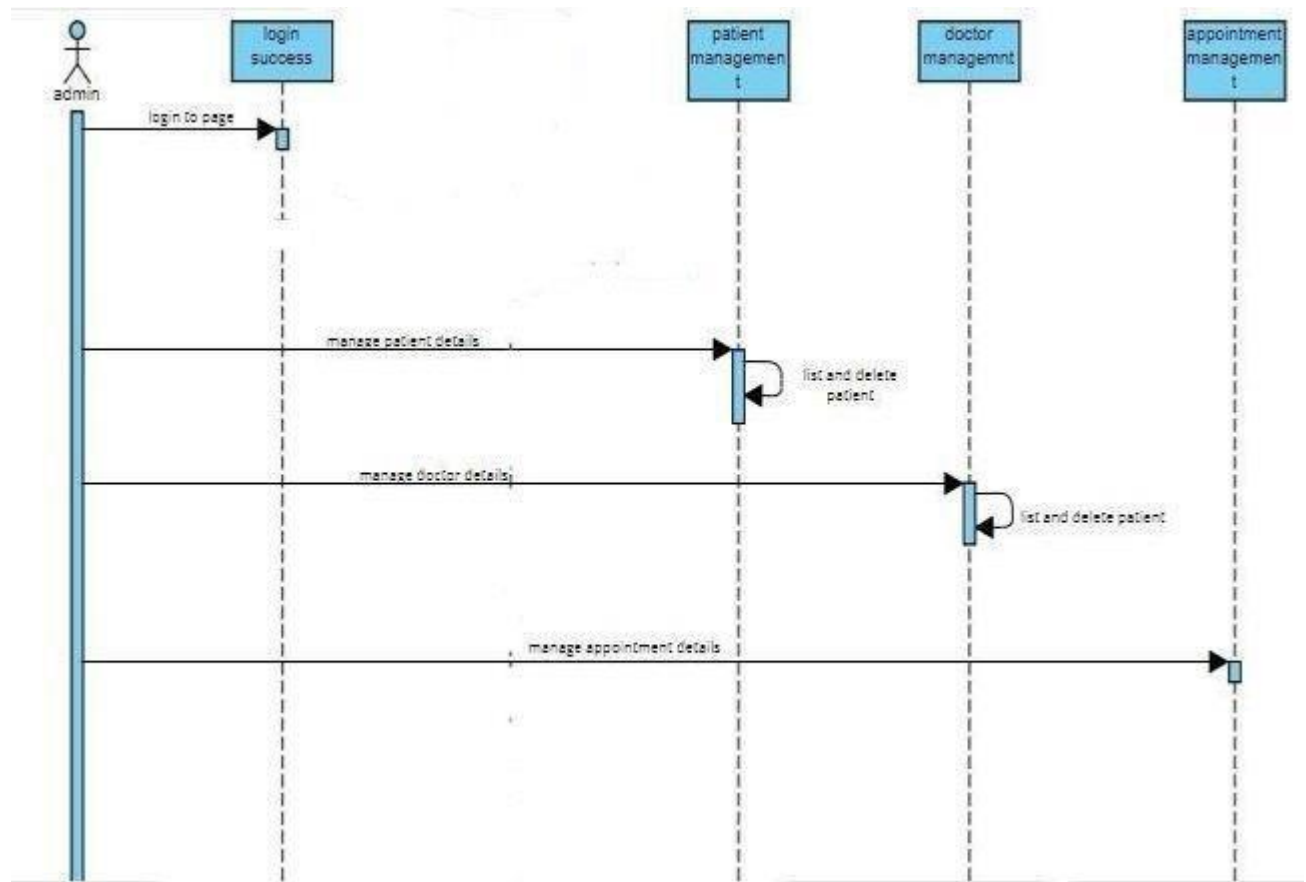
FIGURE: SEQUENCE DIAGRAM

14

5.5 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modeling the activity flow of the system. An application can has multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.
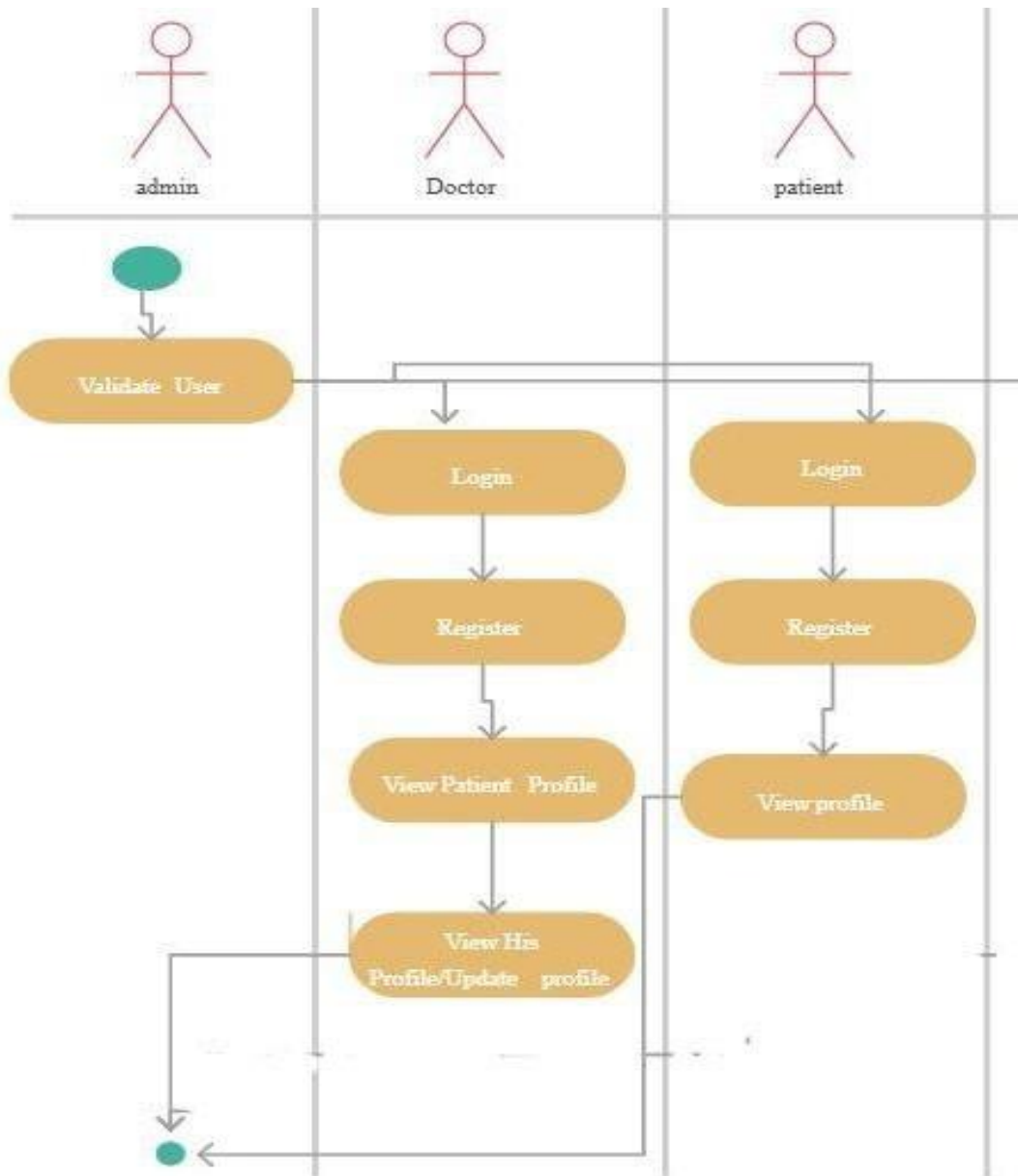
FIGURE: ACTIVITY DIAGRAM

5.7. 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 modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Purpose of Class Diagrams

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
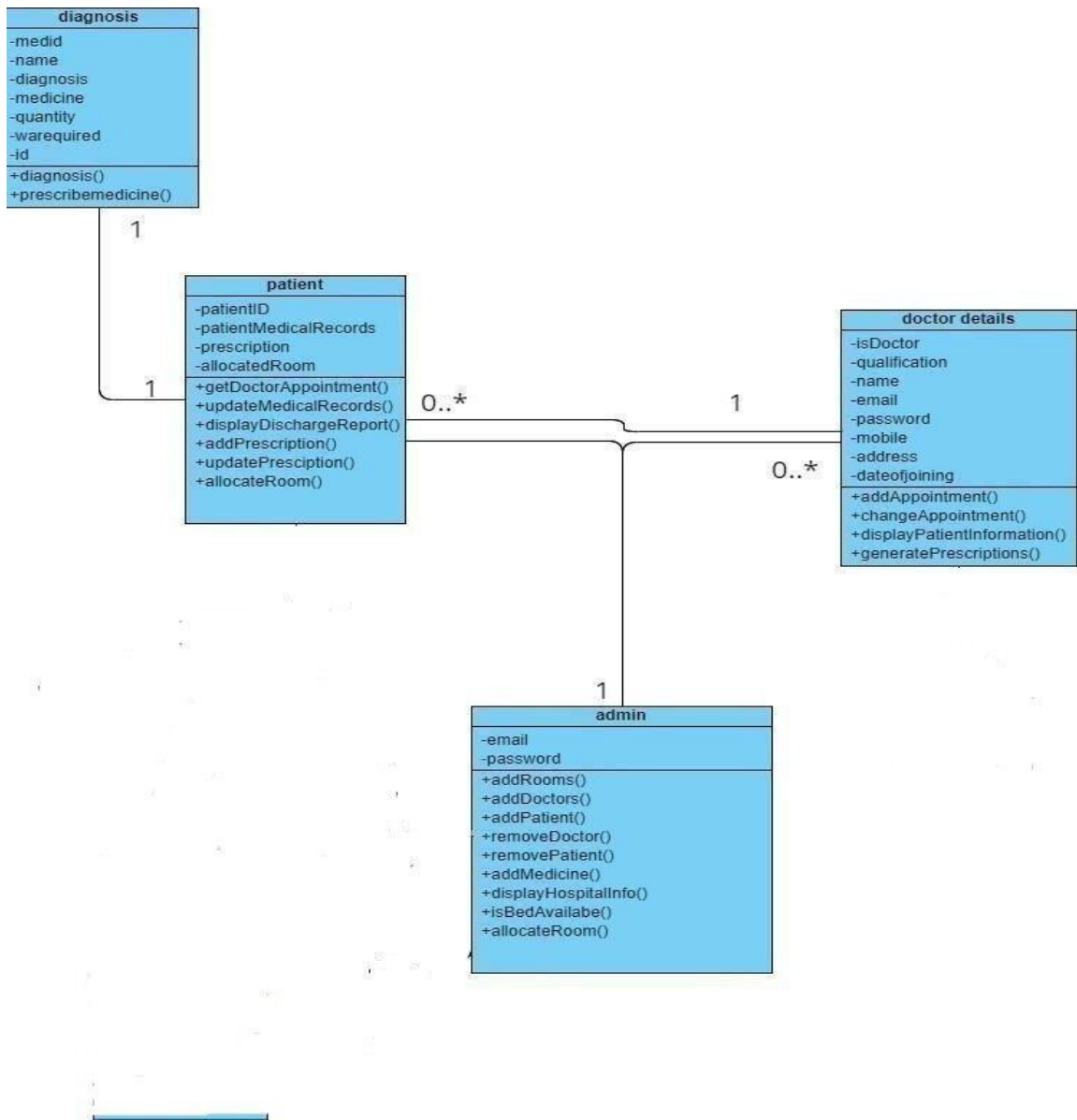
UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

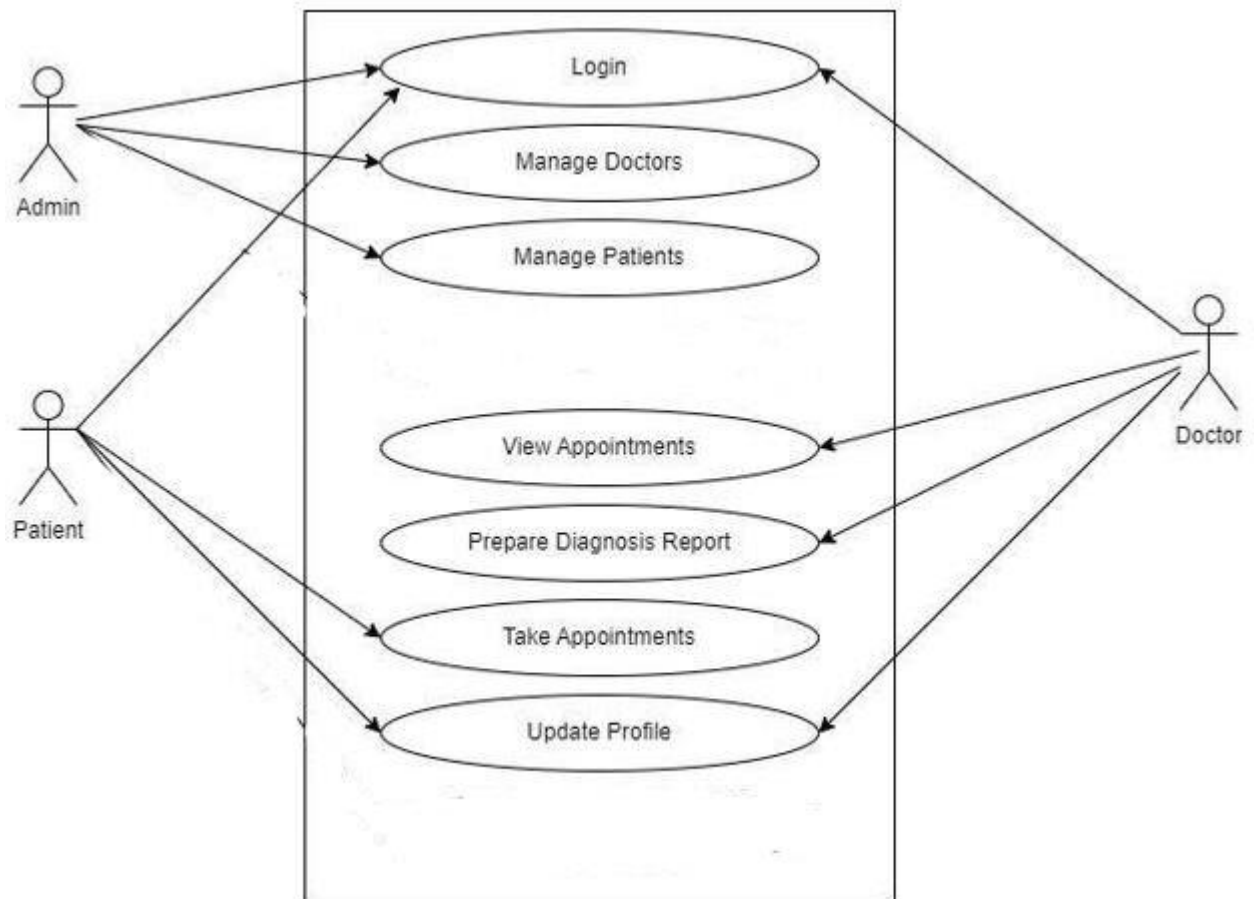**Where to Use Class Diagrams?**

Class diagram is a static diagram and it is used to model the static view of a. The static view describes the vocabulary of the system. Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the system static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception.

Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.

diagnosis
- -medid
- -name
- -diagnosis
- -medicine
- -quantity
- -warequired
- -id
- +diagnosis()
- +prescribemedicine()

1

patient
- -patientID
- -patientMedicalRecords
- -prescription
- -allocatedRoom
- +getDoctorAppointment()
- +updateMedicalRecords()
- +displayDischargeReport()
- +addPrescription()
- +updatePresciption()
- +allocateRoom()

1

0..*

doctor details
- -isDoctor
- -qualification
- -name
- -email
- -password
- -mobile
- -address
- -dateofjoining
- +addAppointment()
- +changeAppointment()
- +displayPatientInformation()
- +generatePrescriptions()

1

0..*

1

admin
- -email
- -password
- +addRooms()
- +addDoctors()
- +addPatient()
- +removeDoctor()
- +removePatient()
- +addMedicine()
- +displayHospitalInfo()
- +isBedAvailabe()
- +allocateRoom()

5.7. USE CASE DIAGRAM

5.8 USER INTERFACE DESIGN

The design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether that be software design, user research, web design, or industrial design.

6 IMPLEMENTATION AND TESTING

Homepage
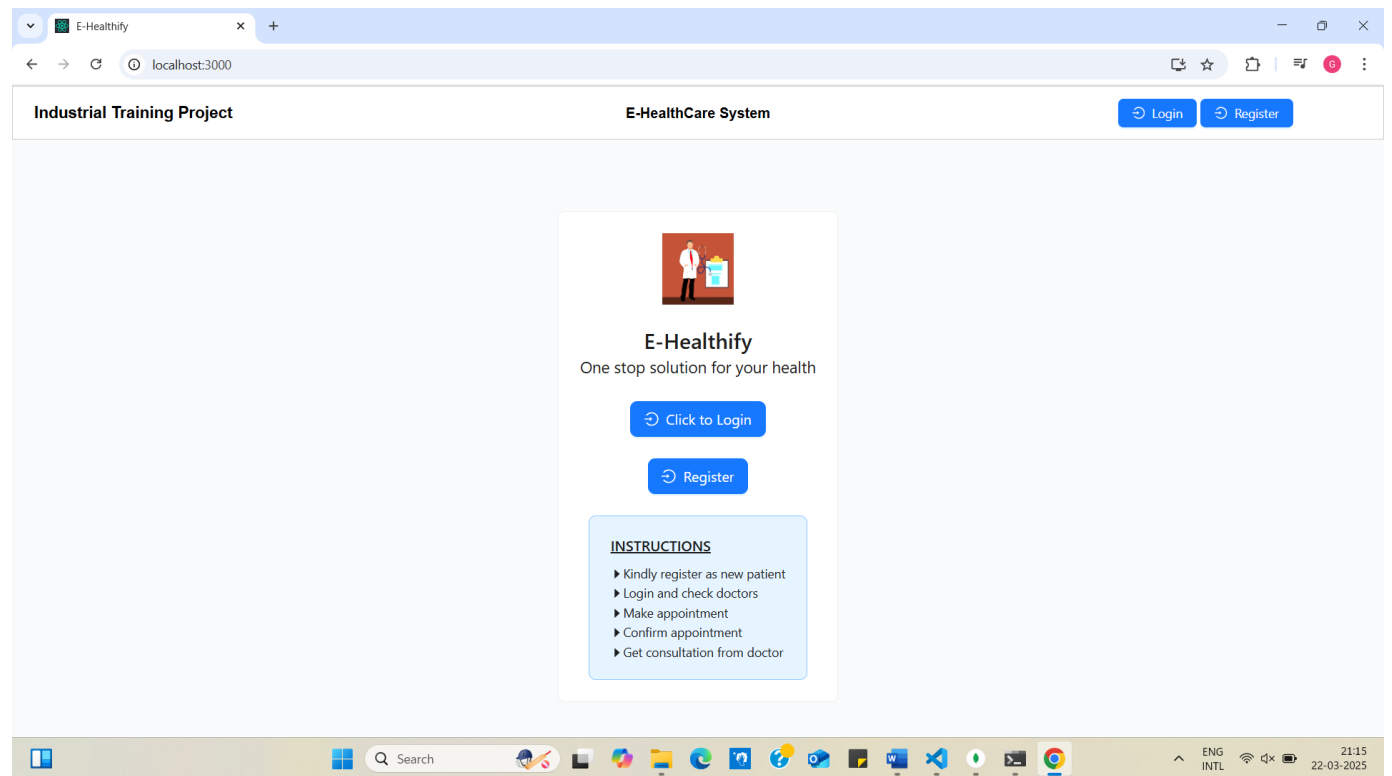
Login page (same page for admin/patient/doctor)
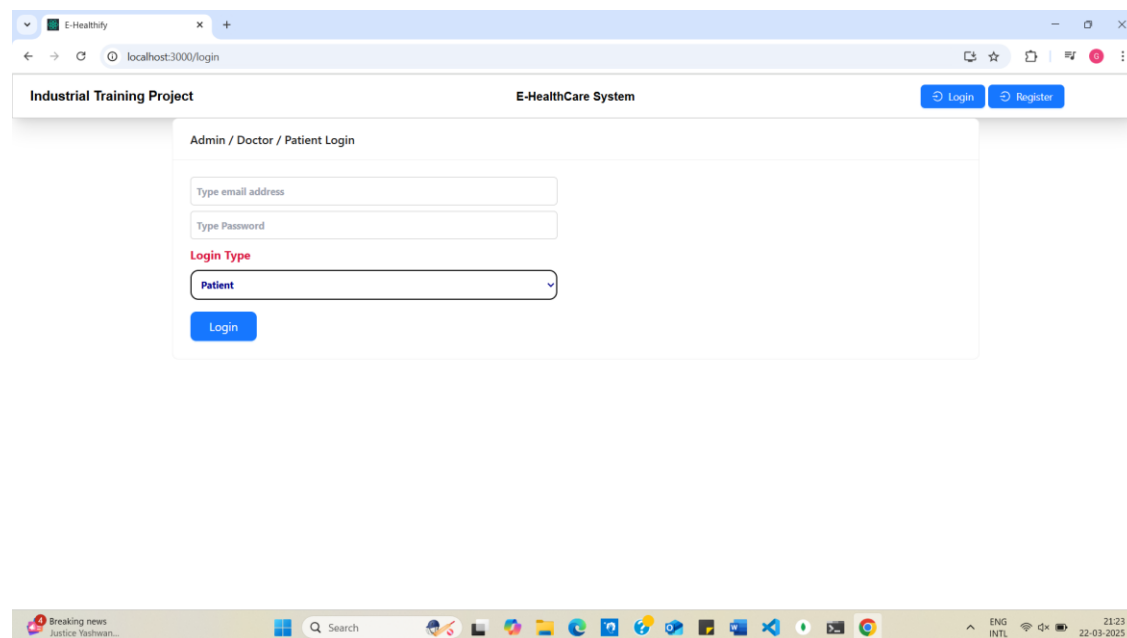
Industrial Training Project   E-HealthCare System   ⟲ Login   ⟲ Register

Admin / Doctor / Patient Login

Type email address

Type Password

Login Type

Patient ⌄

Patient
Doctor
Admin

Password change (common for doctors/ patients login)

⌄   🖼 E-Healthify   ×   +   — ▢ ✕

← → C   ⓘ localhost:3000/cpass   🖵 ☆   🗗 | ☰ 🔴 ⋮

Industrial Training Project   E-HealthCare System   Change Password   Home   Logout

Welcome Shankar G M

Change Password

Enter new Password

Confirm new Password

Save   Go Back

4 29°C
Haze   🔲 Q Search   🖌   ▰ 🔵 📁 🌐 🟠 🟡 🔵 ⬛ 📘 ✖ 🟢 ⬛ 🔵   ∧ ENG 🛜 🔊 🔋   21:22
INTL   22-03-2025

Patient registration

23

Admin login

Add Doctor

Admin login - View all doctors

Patient login

Book an Appointment

Patient login – Find all appointments

Doctor login

View appointments (sorted by date)

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be through enough to be useful but not so through that no one outside the test group will read it.

6.1. Introduction

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error prone areas in a system. Testing must be through well planned. A partially tested system is to detect errors and error prone areas in a system. Testing must be through well planned. A partially tested system is as bad as an untested system. And the price of an untested and under tested system is high.

6.2. Objectives of Testing

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

6.3. Process Overview

The following represents the overall flow of the testing process:

- Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.

- Identify which particular test(s) will be used to test each module.

- Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.

- Identify the expected results for each test.

- Document the test case configuration, test data, and expected results.

- Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).

- Successful unit testing is required before the unit is eligible for component integration/system testing.

- Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, it's possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.

6.4. Test Cases

A test case is a document that describe an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data.

Requirement expected results. The process of developing test cases can help find problems in the requirements or design of an application since it requires completely thinking through the operations of the application.

6.5. Testing Steps

• Unit Testing

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

• Integration Testing

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conductingtest to uncover errors associated within theinterface.

6.6. Validation

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exist.

a) The function or performance characteristics confirm to specification and are accepted.

b) A deviation from specification is uncovered and a deficiency lists is created.

c) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

Tested By:	M GOWRI SHANKAR
Test Type	Unit Testing
Test Case Number	1
Test Case Name	Doctor Login / Patient login / Admin
Test Case Description	The user should enter username and password. The test case will check data from the database and if the username and password is correct it should be redirected to homepage or else it should deny the access.

Item(s) to be tested	
1	Verification of login page

Specifications	
Input	**Expected Output/Result**
1) Enter the correct username and password	1) Login Successful
2) Enter invalid username and password	2) Login Denied

Tested By:	M GOWRI SHANKAR
Test Type	Unit Testing
Test Case Number	2
Test Case Name	Register new patient
Test Case Description	New user registration as patient should allow getting all details of the patient - Patient name, Mobile number, Email, City, Blood Group, Age and on clicking submit should store these details in a separate record in the patients collection.

Item(s) to be tested	
1	Registration of new patient

Specifications	

Input	Expected Output/Result
If all details are correct and Email ID doesn't exist in the collection,	Registration Successful
If mandatory details are missing or duplicate email id exists in patient collection	Registration not successful

Tested By:	M GOWRI SHANKAR
Test Type	Unit Testing
Test Case Number	3
Test Case Name	Admin login
Test Case Description	On login, should be redirected to admin page that contains options to add new doctor, view all doctors

Item(s) to be tested	
1	Add new Doctor, View all doctors

Specifications	
Input	**Expected Output/Result**
On clicking add new doctor, should get doctor details – Doctor Name, Department, Qualification, Designation, Years of Experience.	If mandatory details are filled, doctor should be added to Doctors collection If mandatory fields are missing, adding new doctor will be unsuccessful
On Clicking view all doctors, should display all the doctors sorted by department	If doctors record is present in the collection, it should be displayed as table If no doctors are present, empty table is displayed

Tested By:	M GOWRI SHANKAR
Test Type	Unit Testing
Test Case Number	4
Test Case Name	Patient login
Test Case Description	After patient logins using valid username and password, should be redirected to homepage containing book an appointment and find your appointment options

Item(s) to be tested	
1	Book an appointment, Find an appointment

Specifications	
Input	**Expected Output/Result**
On Clicking Book an Appointment, Selecting date of appointment, entering description of issue and select the doctor from the list of doctors shown.	If all details are provided for appointment, new appointment record with doctor and patient details must be added to Appoints collection.
On Clicking find your appointment, appointments made by this patient (obtained from login record) must be displayed as table	If no appointments are made, empty table is displayed, Or else appointments sorted using date is displayed as a table.

6.7. White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

6.8. Black box testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

6.9. System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity.

6.10. Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hardcopy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

6.11. User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

a)   Input Screen Design.

b)   Output Screen Design.

c)   Format of reports and other outputs

6.12. Integration Testing

Softwaretestingis always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

- Whether the software matches with the specification (i.e. Process base) to verify the product.
- Whether this software in one client what wants (i.e. product base) to validate the product.
- Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behavior if exists. Later all the module was integrated and flat file was generated.

6.13. Functional Testing

These are the points concerned during the stress test:

- Nominal input: character is in putted in the place of digits and the system has to flash the message "Data error"
- Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

7   SYSTEM SECURITY MEASURES

7.1. Database Security

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So, you should follow some security measures. We have used security levels in database level at system level.

7.2. System Security

If we talk about the system security in our proposed system we have implemented with the help of maintain the session throughout the system's use. Once a user has logged out than he/she will not be able to perform any task before signing back again.
A high level of authentic login is given to the system so this is a very tedious task to enter without authorization and authentication.

7.3. Limitation

Complexity

- Awareness of Healthcare advantages & importance.
- In general, Experience and knowledge of using computer applications.
- Impressions and Beliefs regarding Healthcare and making use of them efficiently.
- There is a shortage of professional healthcare faculty with in-depth knowledge of Healthcare and other similar technologies.
- Poor acceptance of Healthcare Software.
- Lack of health informatics professionals capable of establishing and implementing the techniques.

8. COST ESTIMATION

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
2. Determinea set of15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort E_i in person-months the equation used is of the type is shown below

E_i=a*(KDLOC)b

The value of the constant a and b are depending on the project type.

In COCOMO, projects are categorized into three types:

1. Organic
2. Semidetached
3. Embedded

1. Organic: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. **Examples of this type of projects are simple business systems, simple hospital management systems, and data processing systems.**

2. Semidetached: A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. **Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex hospital management system.**

3. Embedded: A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. **For Example:** ATM, Air Traffic control.

For three product categories, Bohem provides a different set of expression to predict effort (in a unit of person month) and development time from the size of estimation in KLOC(Kilo Line of code) efforts estimation takes into account the productivityloss due to holidays, weekly off, coffee breaks, etc.

According to Boehm, software cost estimation should be done through three stages:

1. Basic Model
2. Intermediate Model
3. Detailed Model

1. Basic COCOMO Model: The basic COCOMO model provide an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

Effort=a_1*(KLOC)a_2 PM
Tdev=b_1*(efforts)b_2 Months

Where, **KLOC** is the estimated size of the software product indicate in Kilo Lines of Code,

a_1, a_2, b_1, b_2 are constants for each group of software products,

Tdev is the estimated time to develop the software, expressed in months,

Effort is the total effort required to develop the software product, expressed in **person months (PMs)**.

Estimation of development effort

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic: Effort = 2.4(KLOC) 1.05 PM

Semi-detached: Effort = 3.0(KLOC) 1.12 PM

Embedded: Effort = 3.6(KLOC) 1.20 PM

41

According to Boehm, software cost estimation should be done through three stages:

1. **Basic Model**
2. **Intermediate Model**
3. **Detailed Model**

1. Basic COCOMO Model: The basic COCOMO model provides an accurate size of the project parameters.

The following expressions give the basic COCOMO estimation model:

Effort applied to the project: $E = a_b(KLOC)^{b_b}$ (in Person-month)
Development time: $D = c_b(E)^{d_b}$ (in month)
Manpower required: $P = \frac{E}{D}$ (in Person)
Where a_b, b_b, c_b, d_b are constants for each category of software product

Software Project	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Software project using semi-detached mode with 7 Kloc.

Effort (E) = a*(KLOC)b = 3.0*(0.07)1.12 = 26.523513875326465 PM

Development Time (D) = c*(E)d* = 2.5(0.08155)0.35 = 7.8742 Months(M)

Person Required (P) = E/D = 1784.42/34.35 = 3.36837 Persons ~4 Person

Note: With extra manpower the project is completed within 1 month

9. CONCLUSION

This E health care management system is a standalone application that assists in management of staffs, doctors and patients in easy, comfortable and effective service. Here, the concept of data mining plays vital role to develop an effective health care management system.

The proposed application aims to create a friendly working environment for any health care centers and to overcome the drawbacks in existing system of health care management. This system is very reliable and flexible from all aspects, so new features and modules can be easily integrated into the system in future.

10. FUTURE SCOPE AND FURTHER ENHANCEMENTS

Since this project was started with a good knowledge about the Healthcare Management System, we came to know about the enhancement capability during the process of building it. Some of the scope we can increase for the betterment and effectiveness are listed below:

- Build a complete web application using MVC Framework
- Manage doctor and patient from admin control panel.
- Use of Oracle as its database.
- Online payment system can be added.
- Making the system flexible in any type.
- A separate module for pharmacycan be added in the near future.

11. REFERENCES

- https://www.w3schools.com

- https://www.slideshare.com

- https://www.tutorialspoint.com

- https://www.youtube.com

- Mern: A Full Stack Development [2] Intelligent Hospital Management System - Research Paper By Baki Koyuncu, Hakan Koyuncu.
- Hospital Management System Using Web Technologies – By Reva University.
- Ecommerce Web Application Using Mern Technology
- Hospital Management System – By Digvijay. H. Gadhari, Yadnesh. P. Kadam And Prof.Parineeta Suman.
- Rfid Based Smart Hospital Management System – Ieee Research Paper.
- Study Of Advanced Hospital Management System -- By Anna University.
- Hospital Management System – By Prajakta Musale, Aryan Pokharkar, Apoorva Popalghat, Akhilesh Poke, Abhishek Pote And Harsh Pokharna
- Internet Articles From Websites – Geeks For Geeks, Stackoverflow, Etc. And Videos From Internet.