A PROJECT REPORT ON.

**HMS : Hospital Management System**

SUBMITTED BY:

Shifa Hamid Ansari

Roll No**: 109**

Seat No**: 927**

SUBMITTED TO:

**M.C.E SOCIETY’S**

# **ALLANA INSTITUTE OF MANAGEMENT**

**SCIENCES, PUNE**

2022-2024

UNDER THE GUIDANCE OF:

**Prof. Pankaj Kulkarni**

### ACKNOWLEDGEMENT

I take the opportunity to thank all those who have been involved in this project one way or the other.

Firstly, I thank God, Almighty for blessing us with his grace and taking our endeavour to a successful culmination.

It is my prime duty to offer my gratitude to Savitribai Phule Pune University for including the project work in the syllabus of First Year

Master’s degree so as to develop interest about research work among the student like us.

I wish to express my sincere thanks to Prof. Pankaj Kulkarni. sir for giving me the opportunity to complete the project work by providing facility in the department and providing valuable guidance to complete the task.

I am greatly obliged to Mrs. Tabassum Ma’am Head of Department of MCA who provided valuable guidelines and conceptual guidance throughout the project work also helped out in clearing concept about project.

I also grateful to Mr. Roshan Kazi Sir, Director in-charge of Allana Institute of Management Sciences for providing all necessary facilities of laboratory and library at college.

Last but not the least my special thanks to my parents, my friends and also those people who have encouraged me, helped me to complete my project successfully in time.

## INTRODUCTION

Our project Hospital Management system includes registration of patients, storing their details into the system, and also booking their appointments with doctors.

Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically.

User can search availability of a doctor and the details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or patient.

Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast

It is having mainly three modules. One is at Administration Level and other two are at user i.e., of patients and doctors. The Application maintains authentication in order to access the application.

Administrator task includes managing doctors’ information, patient’s information. To achieve this aim a database was designed one for the patient and other for the doctors which the admin can access. The Patient modules include checking appointments, prescription.

Hospital Management System In PHP is web based application.

The project Hospital Management system includes registration of patients, storing their details into the system. The software has the facility to give a unique id for every patient and stores the details of every patient.

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

## PROBLEM DEFINATION

In this busy world we don’t have the time to wait in infamously long hospital queues. The problem is, queuing at hospital is often managed manually by administrative staff, then take a token there and then wait for our turn then ask for the doctor and the most frustrating thing - we went there by traveling a long distance and then we come to know the doctor is on leave or the doctor can’t take appointments. Medipro will help us overcome all these problems because now patients can book their appointments at home, they can check whether the doctor they want to meet is available or not. Doctors can also confirm or decline appointments, this help both patient and the doctor because if the doctor declines’ appointment, then patient will know this in advance and patient will visit hospital only when the doctor confirms’ the appointment this will save time and money of the patient. Hospital management system is essential for all healthcare establishments, be it hospitals, nursing homes, health clinics, rehabilitation centers, dispensaries, or clinics. The main goal is to computerize all the details regarding the patient and the hospital. The installation of this healthcare software results in improvement in administrative functions and hence better patient care, which is the prime focus of any healthcare unit.

Benefits of implementing a hospital management system:

• Appointment booking - Helps patients cut the long queue and saves their time

• Role-Based Access Control - Allows employees to access only the necessary information to effectively perform their job duties. Increases data security and integrity

• Overall cost reduction - Cuts down paper costs as all the data are computerized. No separate costs for setting up physical servers

• Data accuracy - Removes human errors

• Data security - Helps to keep patients records private. Restricts access through role-based access control

• Revenue management - Makes daily auditing simple , helps with statistics and other financial aspects

## NEED OF COMPUTERIZATION

Since the existing system for maintaining the data in not efficient, it needs to be made efficient. In existing system hospital keeps the tracking records in the registers and files. In this, if doctor has to search for any specific appointments, then doctor has to go through whole registers and files. It takes too much time. Also when one treatment is completed, doctor has to remove his data or somehow remove his entry.

At some places some hospitals uses Microsoft Excel for keeping a track of records, but in this system also, searching and updating is time consuming. So to overcome above drawbacks, new easier and less time consuming system is needed which will give easy way to maintain track of records.

Hospitals can also track status of patients easily.

## SCOPE OF PROPOSED SYSTEM

This system will be designed for local level and city level. So the system can be used for many doctors at the hospital. The data in this will be secured. The doctors can find, edit and update data of patients by simple methods.

The data access will be easier and faster. Hospitals can view pending appointments/patients very easily. And the data update can be made only by the admin/doctor.

This new system will be easier and less time-consuming system. Also, this System will give easy way to maintain track of patients and appointments.

### OBJECTIVES OF THE PROPOSED SYSTEM

* The main objective of the system is to lessen the human efforts.
* Besides this data security is also an objective.
* Maintaining correct information is easy using the application.
* Data cannot be stolen or lost.

## FACT FINDING TECHNIQUE

To study any system the analysts needs to collect facts and relevant information. The facts when expressed in quantitative form are termed as data. For getting information and data we have used the following techniques

1. **Interview**

It is used to collect information from group or an individual. In the interview the analysts sit face to face with the interviewee. The information in this technique is accurate and reliable as the analyst can clear and cross check the doubts there itself.

### Guidelines to be kept in mind

Prepare notes in advance

* Avoid any arguments
* Be a very good listener
* Evaluate the outcome of the interview

In our interview we asked the following questions to the manager of the company.

* What is Hospital Management?
* How does it work?
* What types of data are recorded in the database?
* What is current system being used by owner and what are it advantages and disadvantages?
* Is the present system efficient?
* What would be the system that the owner would like to use?
* Will this system be cost effective and what will be its advantages?
* How will this system help the owner?
* Will this system be secured?

2. **Questionnaire**

The aim of questionnaire is to gather information for *purpose* of analysis. The questions are simple and to the point it is the cheapest and the most feasible way of gathering data.

**Guidelines**

* + Must have a specific objective.
  + Questions must be in simple language.
  + It must be straight to the point.

The following questions were asked to the owner of this business.

* Is the current system effective?
* What changes would you like to make to the present system?
* What are the disadvantages of the present system?
* Which new system would you prefer?
* What will be its advantages?
* Will the new system be effective for the business?
* Will it be easy to use by the other people in warehouse?
* What will be its short comings?

**1.Record View**

For more information we had referred to some old documents and excel files from the owner.

### SOFTWARE & HARDWARE REQUIREMENT

* **Operating System***:* Windows 10, Windows XP, and Linux
* **Database***:* MySQL
* **Microprocessor***:* Pentium 4 or Higher
* **RAM***:* 512 Mb or Higher
* **HDD***:* 50 Mb+
* **Other Requirements***:* MySQL Connector
* **Language used** : PHP

**ANALYSIS SPECIFICATION**

**Class Diagram:**

Class:

A class is a description of a set of objects that share the same attributes, operations, relationships and semantics. Names: Name must be singular. Every class must have a name that distinguishes it from other classes. A name is a textual string.

Attributes: Attribute is a named property of a class that describes a range of values that instances of a property may hold. A class may have any numbers of attributes or no attribute at all.

|  |
| --- |
|  |
|  |
|  |

Operations: An operation is an implementation of a service that can be requested from any object of the class. An operation is an abstraction of something that you can do to an object and that is shared by all objects of that class.

|  |
| --- |
|  |
|  |
|  |

**Use Case Diagram:**

Use Case: A use case specifies the behaviour of a system or a part of a system and is a description of a set of sequence of actions including variants that a system performs to yield a result of a value to an actor.

Actor: The actor represents a coherent set of roles that users f use case play when interacting with these use cases. Actor can be human or they can be automated systems, hardware device.

Use case: Every use case must have a name that distinguishes it from other use cases. A name is a textual string.

System Boundary: It is optional and shown as a rectangle system boundary distinguishes ‘what’ is inside from what is outside the system

**Activity Diagram:**

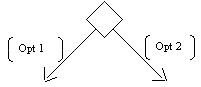
Initial Activity: This shows the starting point or first activity of the flow. Denoted by a solid circle. This is similar to the notation used for Initial State.



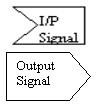
Activity: Represented by a rectangle with rounded (almost oval) edges

.

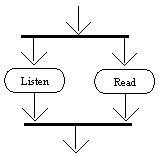
Decisions: Similar to flowcharts, a logic where a decision is to be made is depicted by a diamond, with the options written on either sides of the arrows emerging from the diamond, within box brackets.



Signal: When an activity sends or receives a message, that activity is called a signal. Signals are of two types: Input signal (Message receiving activity) shown by a concave polygon and Output signal (Message sending activity) shown by a convex polygon.



Concurrent Activities: Some activities occur simultaneously or in parallel. Such activities are called concurrent activities. For example, listening to the lecturer and looking at the blackboard is a parallel activity. This is represented by a horizontal split (thick dark line) and the two concurrent activities next to each other, and the horizontal line again to show the end of the parallel activity.



Final Activity: The end of the Activity diagram is shown by a bull's eye symbol, also called as a final activity.

.

**State chart diagram**

Initial State: This shows the starting point or first activity of the flow. Denoted by a solid circle. This is also called as a "pseudo state," where the state has no variables describing it further and no activities.



State: Represents the state of object at an instant of time. In a state diagram, there will be multiple of such symbols, one for each state of the Object we are discussing. Denoted by a rectangle with rounded corners and compartments (such as a class with rounded corners to denote an Object).

We will describe this symbol in detail a little later.



Transition: An arrow indicating the Object to transition from one state to the other. The actual trigger event and action causing the transition are written beside the arrow, separated by a slash. Transitions that occur because the state completed an activity are called "triggerless" transitions. If an event has to occur after the completion of some event or action, the event or action is called the guard condition. The transition takes place after the guard condition occurs. This guard condition/event/action is depicted by square brackets around the description of the event/action (in other words, in the form of a Boolean expression).



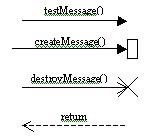
History States: A flow may require that the object go into a trance, or wait state, and on the occurrence of a certain event, go back to the state it was in when it went into a wait state—its last active state. This is shown in a State diagram with the help of a letter H enclosed within a circle.



Event and Action: A trigger that causes a transition to occur is called as an event or action. Every transition need not occur due to the occurrence of an event or action directly related to the state that transitioned from one state to another. As described above, an event/action is written above a transition that it causes.



Sequence diagram, you can represent simple messages, special messages to create or destroy objects, and message responses.



**Component diagram**

Component: The objects interacting with each other in the system. Depicted by a rectangle with the name of the object in it, preceded by a colon and underlined.



Class/Interface/Object: Similar to the notations used in class and object

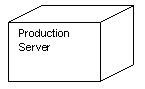
diagrams

Relation/Association: Similar to the relation/association used in class

diagrams

**Deployment diagram**

Node: The element that provides the execution environment for the components of a system. Depicted by a cube with the name of the object in it, preceded by a colon, and underlined.



Connection: Similar to the relation/association used in class diagrams to

define the interconnection between nodes.

**Collaboration diagram.**

Object: The objects interacting with each other in the system. Depicted by a rectangle with the name of the object in it, preceded by a colon and underlined.



Relation/Association: A link connecting the associated objects. Qualifiers can be placed on either end of the association to depict cardinality.

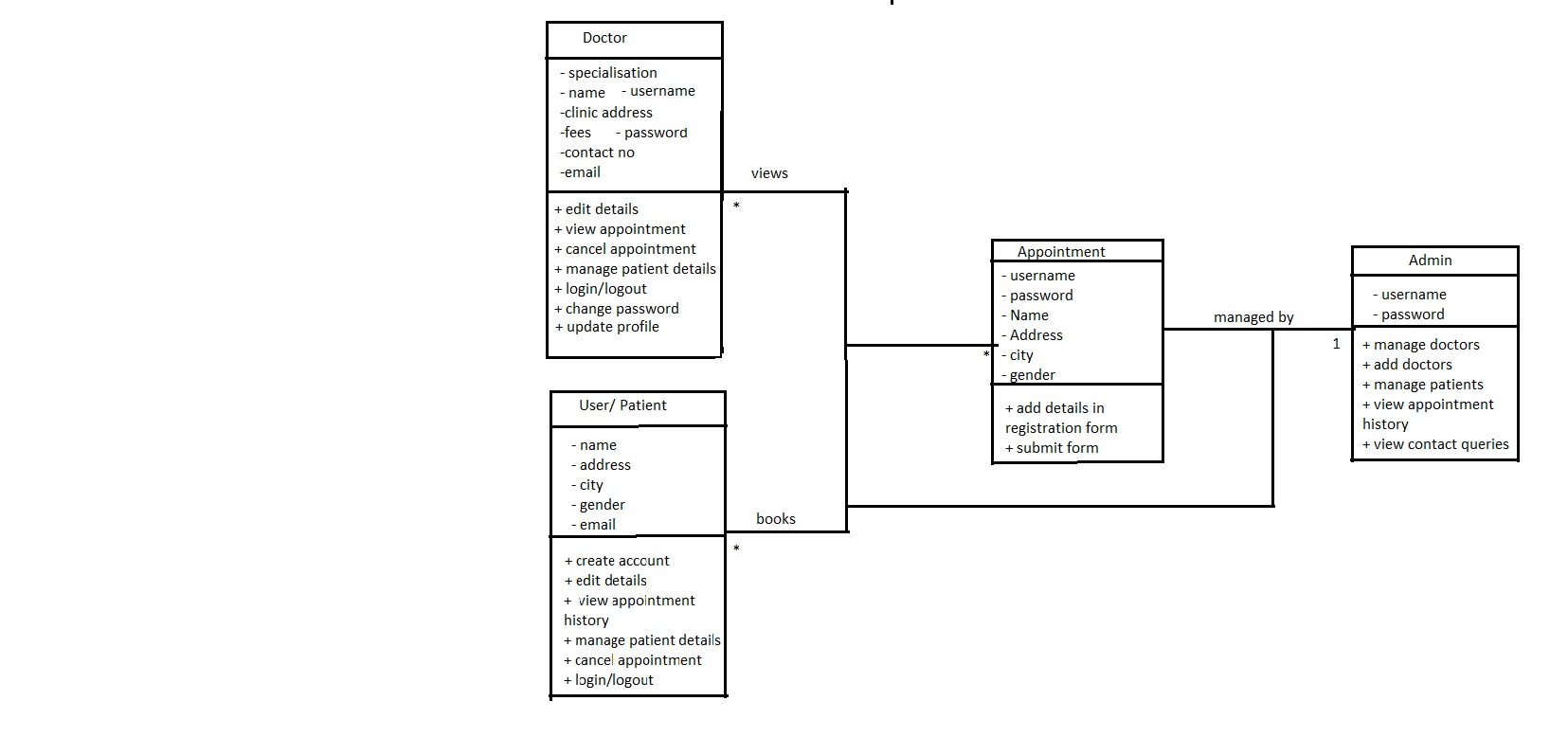


Messages: An arrow pointing from the commencing object to the destination object shows the interaction between the objects. The number represents the order/sequence of this interaction.

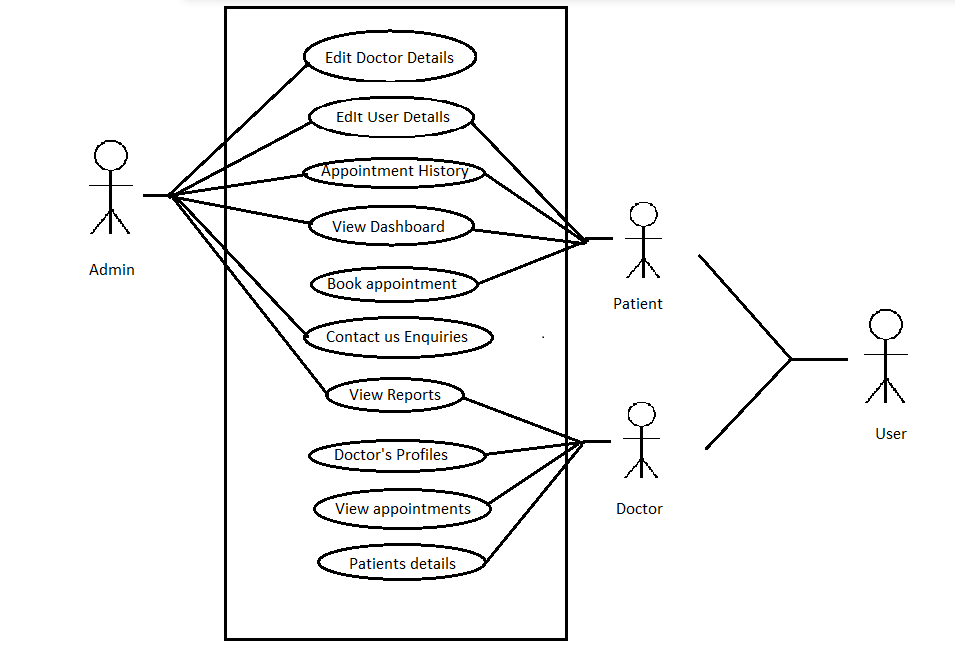


## DESIGN SPECIFICATION

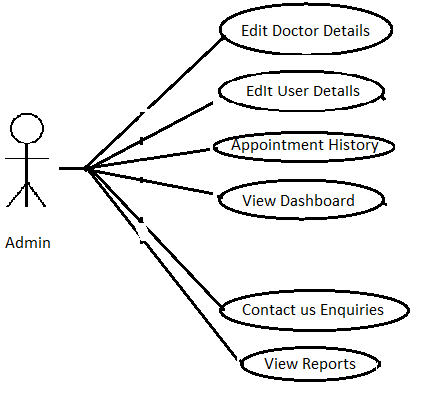
### CLASS DIAGRAM



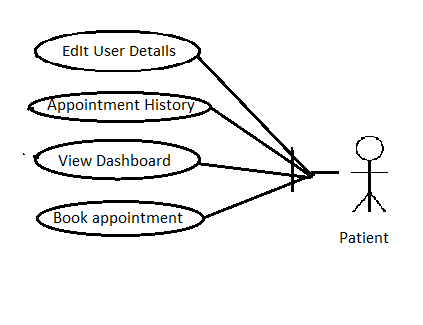
**USECASE DIAGRAM**



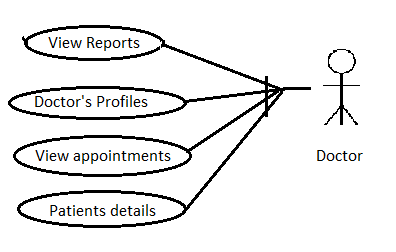
**Admin**



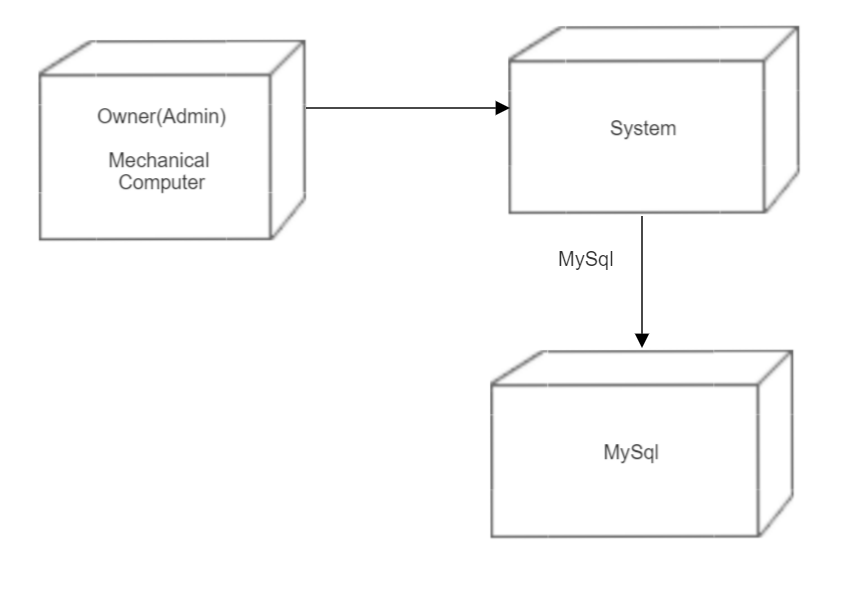
**Patient**



**Doctor**

****

**Deployment Diagram**

****

**DECISION TOOLS**

### ACTIVITY DIAGRAM

|  |  |
| --- | --- |
| Initial Activity |  |
| Activity |  |
| Decisions |  |
| Signal |  |
| Concurrent  Activities |  |
| Final Activity |  |

### STATE CHART DIAGRAM

|  |  |
| --- | --- |
| Initial  State |  |
| State |  |
| Transition |  |
| History States |  |
| Event and  Action |  |
| Signal |  |
| Final  State |  |

### SEQUENCE DIAGRAM

|  |  |
| --- | --- |
| Object |  |
| Message |  |

### COMPONENT DIAGRAM

|  |  |
| --- | --- |
| Component |  |
| Class/interface/Object |  |
| Relation/Association |  |

### DEPLOYEMENT DIAGRAM

|  |  |
| --- | --- |
| Node |  |
| Connection |  |

### COLLABORATION DIAGRAM

|  |  |
| --- | --- |
| Object |  |
| Relation/Association |  |
| Message |  |

### CLASS DIAGRAM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class |  |  | | |  |
|  | | |
| Attribute |  | | | | |
|  | |  |  | |
|  |
|  |
| Operation |  | | | | |
|  | |  |  | |
|  |
|  |

### USE CASE DIAGRAM

Actor

Use

case

System

boundary

## TESTING STRATERGIES

**Simply stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.**

In many software engineering methodologies, the testing phase is a separate phase which is performed by a different team after the implementation is completed. There is merit in this approach; it is hard to see one's own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times. Unfortunately, delegating testing to another team leads to a slack attitude regarding quality by the implementation team.

* **Comparison testing**oComparison of product strengths and weaknesses with previous versions or other similar products. In the previous version the Exchange rate had to be updated manually, Data were stored in Records and employees would make error while noting down customer details. We have overcome this problem by updating rates directly from the internet which fetches the freshest rates from the market and all records are now stored in a Data-Base and Error made by employees can be reduced.
* **Alpha testing**oIn house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing. In Alpha Testing we ourselves have tested the software.
* **Beta testing**

Testing typically done by end-users or others. Final testing is done before releasing the application for commercial purpose. For Beta Testing we have given it to the employees of the Exchange House and have taken their views about the System.

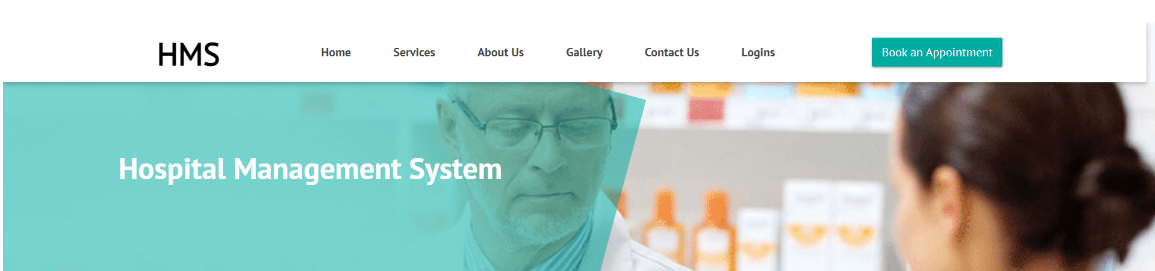
* **Performance testing**oTerm often used interchangeably with ‘stress’ and ‘load’ testing. To check whether system meets performance requirements. We haveused different performance and loading tools to do this.
* **Functional testing**oThis type of testing ignores the internal parts and focus on the output is as per the user requirement or not.
* **Unit testing**oTesting of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test

driver modules or test harnesses.

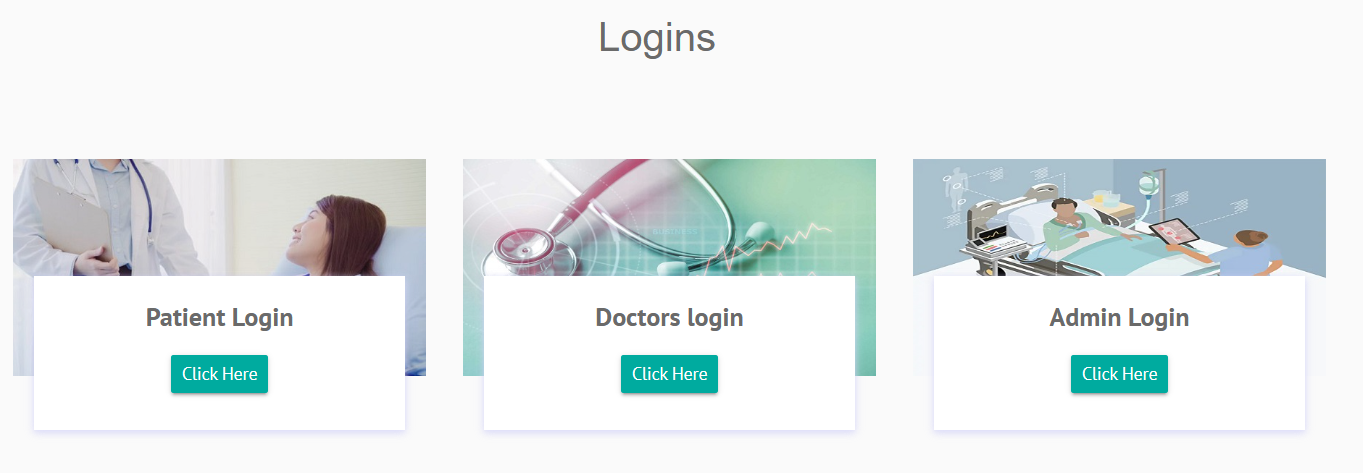
* **Black box testing**oInternal system design is not considered in this type of testing. Tests are based on requirements and functionality.

### I/O Screen

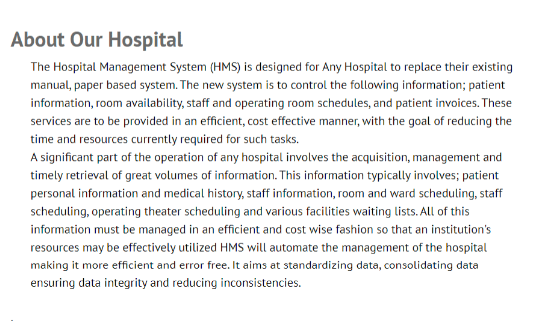
**Dashboard**



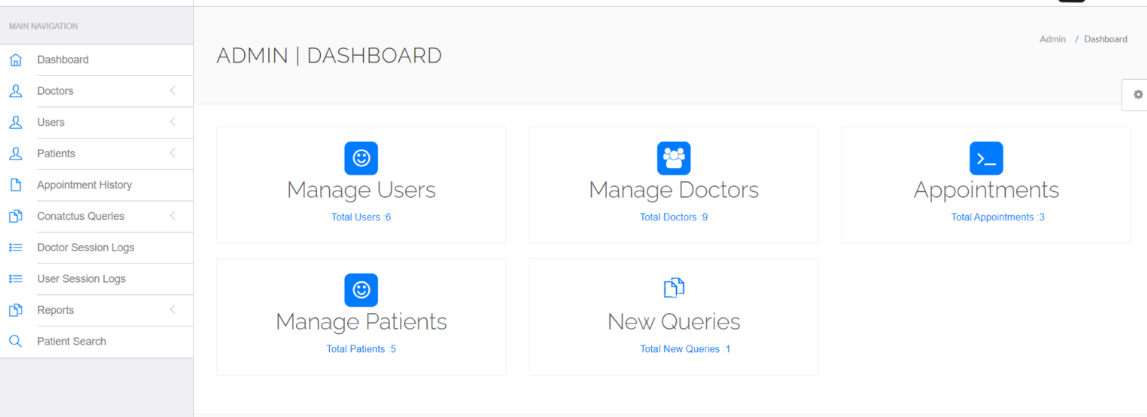
**logins**

****

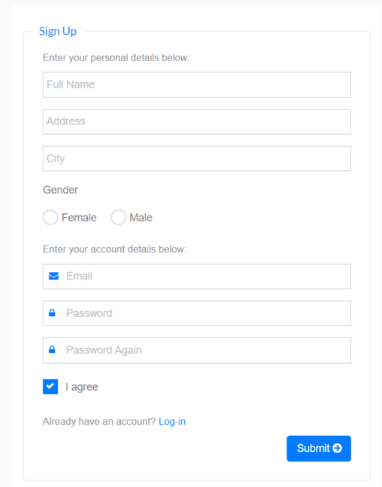
**About Us**

****

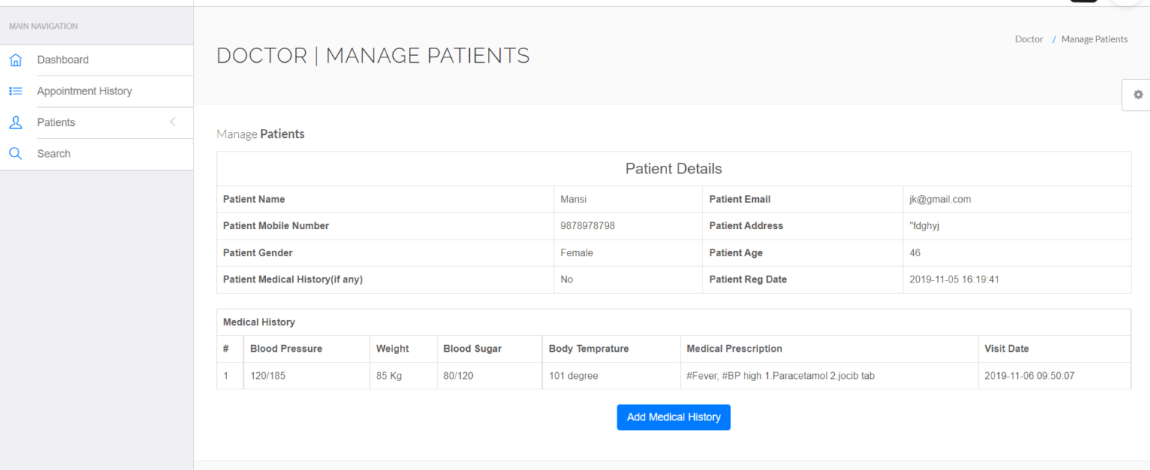
**Admin Dashboard**

****

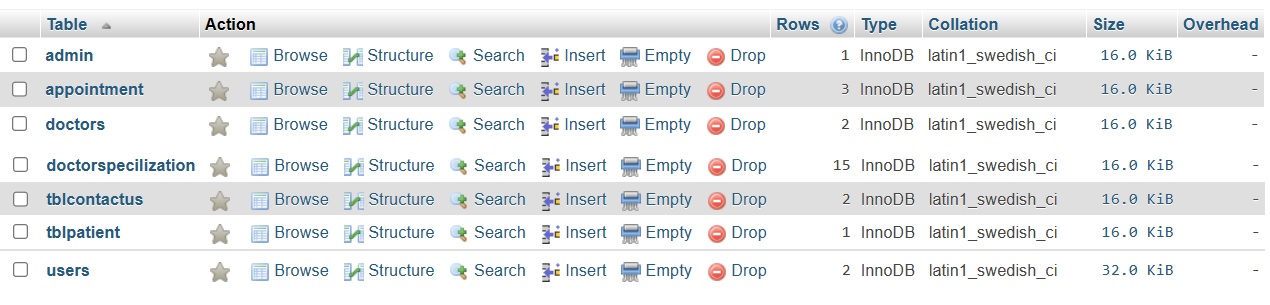
**Patient Registration**

****

**Patient Medical History**

****

### Database



### LIMITATIONS & DRAWBACKS

* This System is designed for small network administration.
* Training every new users of the system is not feasible.
* This system is compatible only with Linux and Windows Operating System Respectively.

### FUTURE ENHANCEMENT

* We can improve functionality of this project by modifying it to make it applicable to run on any operating system.
* Improve flexibility by using advance frontend technique.

### CONCLUSION

### Working on the project was an excellent experience. It helped us to understand the

### importance of planning, designing and implementation so far we have learnt in our

### theory books. It helped us unleashing our creativity while working in a team. It also

### realized the importance of team working, communication as a part of this project.

### The project was successfully completed after a lot of efforts and work hours. This

### project underwent number of compiling, debugging, removing errors, making it bug

### free, adding more facilities in Hospital Management System and interactivity making it more reliable and useful.

There are also few features which can be integrated with this system to make it more flexible. Below list shows the future points to be consider :

• Including a different module for pharmacy, LAB, Bed Allotment and many more.

• Including a Frequently Asked Questions Section.

### BIBLIOGRAPHY

# PHP: A BEGINNER'S GUIDE by Vikram Vaswani

### HTML and CSS: Design and Build Websites, by Jon Duckett

* [GeeksforGeeks | A computer science portal for geeks](https://www.geeksforgeeks.org/)
* [Online Tutorials Library (tutorialspoint.com)](https://www.tutorialspoint.com/index.htm)
* [PHP: Hypertext Preprocessor](https://www.php.net/)
* [CodePen: Online Code Editor and Front End Web Developer Community](https://codepen.io/)