

1.1 Introduction:

Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, part-by-part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e. Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man

1.2 Overview:

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “Administration support system for medical institutions”.

Problem definition:

- The manual handling of the record is time consuming and highly prone to error.
- keeping track of all the activities and their records on paper is very cumbersome and error prone.
- Recording and maintaining all these records is highly unreliable, inefficient and error-prone.
- It is also not economically & technically feasible to maintain these records on paper.

Proposed solution :

- providing low-cost reliable automation of the existing systems
- The main aim of our project is to provide a paper-less hospital up to 90%
- provides excellent security of data at every level of user-system interaction
- provides robust & reliable storage and backup facilities.

Request Analysis

The request was studied carefully to access what are exactly required. The focus being on the task of determining precisely what originator wants.

The analysis includes as what will be the fields, tables the tools to be used as it is a detailed study of the various operations performed by a system and their relationship within and outside the system. As it points question of what must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis data are collected on the available files, decision points and transaction handled by the present system.

The project is being analyzed with every great care possible.

3. Feasibility Study

After doing the project Hospital management system, study and analyzing all the existing or required functionalities of the system, the next test is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

Economical Feasibility:

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

Technical Feasibility:

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provide in the system, as described in the System Requirement Specification(SRS), and checked if everything was possible using different type of frontend and backend platform.

Technical feasibility centers on the existing computer system (hardware, software etc.)and to what extent it can support the proposed addition. This phase involves financial considerations to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible. Our project is technical feasible as the technologies required are already available software required are VB, MS Access. Hardware requirement are 2GB RAM, 4GB storage Pentium IV processor correct.

The technical issue usually raised during the feasibility stage of the investigation include the followings:

- Does the necessary technology exist to do what is suggested?

- Do the proposed equipment have the technical capacity to hold the data required to use the new system?
 - Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
 - Can the system be upgraded if developed?
- Are there technical guarantees of accuracy reliability case of access and data security?

Operational Feasibility:

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management form the users?
- Will the system be used and work properly if it is being developed and implemented ?

Will there be any resistance from the user that will undermine the possible application benefits?

Behavioral Feasibility:

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user is likely to have towards the development of a system.

Time Feasibility:

Time feasibility check whether the project can be completed within the time frame suggest by customer. As per this project was assigned for four months duration and also study suggest that this much time is enough to complete the project therefore the project is also feasible as far as time is considered.

4. Requirement Gathering

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of the system engineering are refined by establishing and completing information description, a detailed functional description, a representation of system behavior, an indication of performance requirement and design constraints appropriate of validation criteria. The introduction of the software requirement specification states the goals and objectives of the software, describing the context of the computer base system.

RESOURCE REQUIRED

HARDWARE SPECIFICATION

RAM: 1GB or more.

Processor: i3 processor or above.

Hard-Disk: 2GB or more.

System Type: 64-bit Operating System.

SOFTWARE SPECIFICATION

Operating System: Windows family.

Web Technology: PHP, HTML, CSS.

Database:- MySQL5.0.

Server: Wamp Server.

Design Tool: Sublime Text3.

Documentation Tool: Microsoft PowerPoint.

UML Diagram: Star UML.

TECHNOLOGY USED

PHP:

PHP stands for PHP: Hypertext Preprocessor. PHP is a server-side scripting language, like ASP. PHP supports many databases (MySQL, Informix, Mysql, Sybase, Solid, PostgreSQL, Generic ODBC, etc.). PHP is an open source software. PHP is free to download and use. PHP sits between your browser and the web server. When you type in the URL of a PHP website in your browser, your browser sends out a request to the web server. PHP is a server-side, cross-platform, HTML-embedded scripting language. There are over half a million domains running PHP and it is freely available for download.

HTML:

HTML stands for Hyper Text Markup Language, which is most widely used language on web to develop web pages. HTML refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page is called Hypertext.

CSS:

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Tools Used:

MySQL is an open source RDBMS that relies on SQL for processing the data in database. MySQL provides APIs for the languages like C, C++, JAVA, PHP and Python. Users can quickly and easily create and access their fir in a secure and easy way. MySQL is most commonly used for web applications and for embedded applications and has become a popular alternative to proprietary database system because of its speed and reliability. MySQL can run on UNIX, Windows and Mac OS.

DATA FLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notation Yourdon,gane and sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose . The development of DFD's is done in several level . Each process in lower level diagram can be broken down into a more detailed DFD in the next level . The lop-level diagram is often called context diagram . It consist a single process bit, which plays vital role in studying the current system . The process in the context level diagram is exploded into other process at the first level DFD to understand the process.

Larry the idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst Constantine first developed the DFD as a way of expressing system requirement in a graphical from this lead to the modular design.

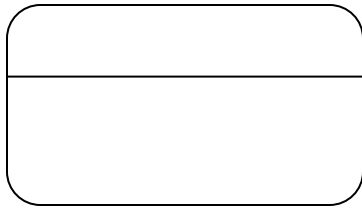
A DFD is also known as a “bubble chart ” has the purpose of clarifying system requirement and identifying major transformation that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consist of a series bubbles joined by data flow in the system.

DFD SYMBOLS:

In the DFD, there are four symbols:

1. A square defines a source or destination of system data.

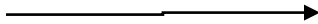
2. An arrow identifies data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.



Forms data flow



Source or destination of



Data flow

Data Flow

1. Data flow has only direction of flow between symbols . It may flow in both directions between a process and a data store to show read before an update .The latter is usually indicated however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. Data flow cannot go directly back to the same process it leads . There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4. A Data flow to a data store means update (delete or change).
5. A data flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Flow charts are easy to understand diagram showing how steps in a process fit together . This makes them useful tools for communicating how processes work , and for clearly documenting how a particular job is done. Furthermore , the act of mapping a process out in flow chart format helps you cleaify your understanding of the process , and helps you think about where the process can be improved. A flow chat therefore be used to:

- Define and analyze processes.
- Build a step by step picture of the process for analysis , discussion, or communication .
- Define standardize or find areas for improvement in a process.

How to use the tool:

Most flow charts are made up of three main types of symbol:

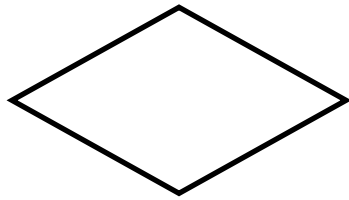
- Elongated circles, which signify the start or end of a process.



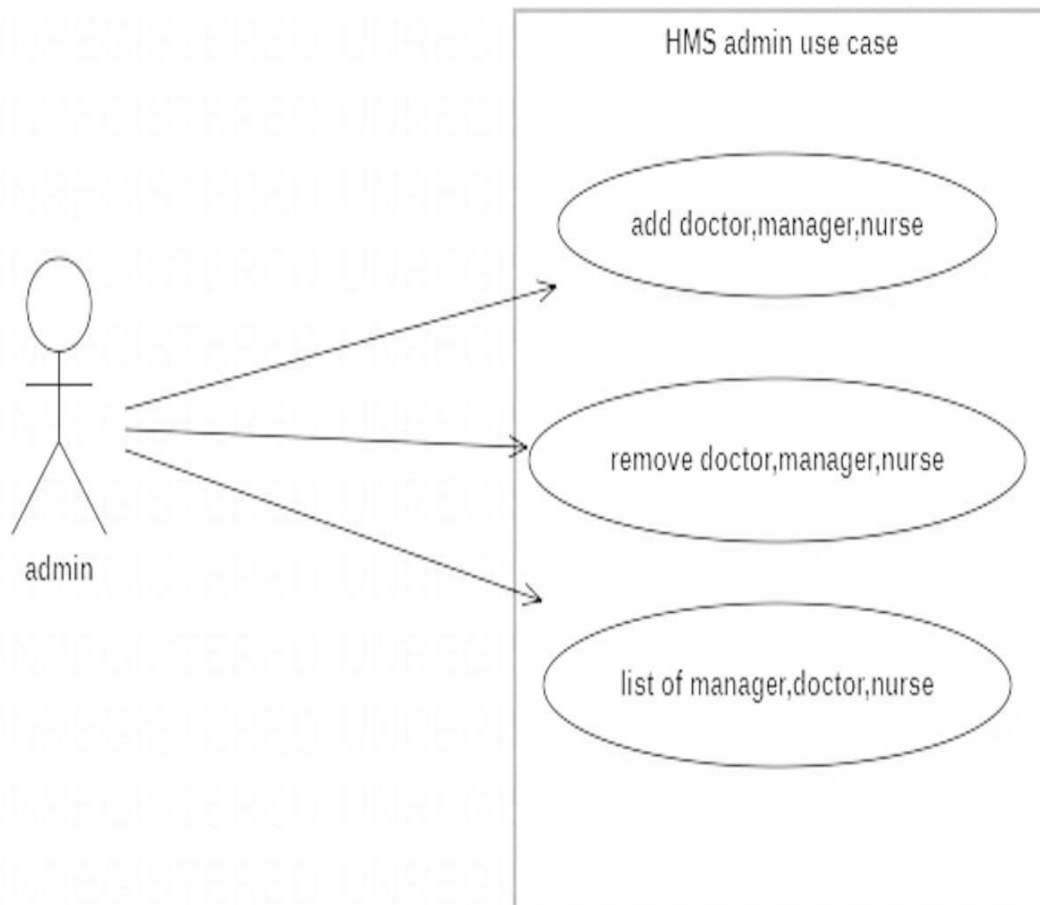
- Rectangles , which show instructions or actions



- Diamonds , which show decision that must be made

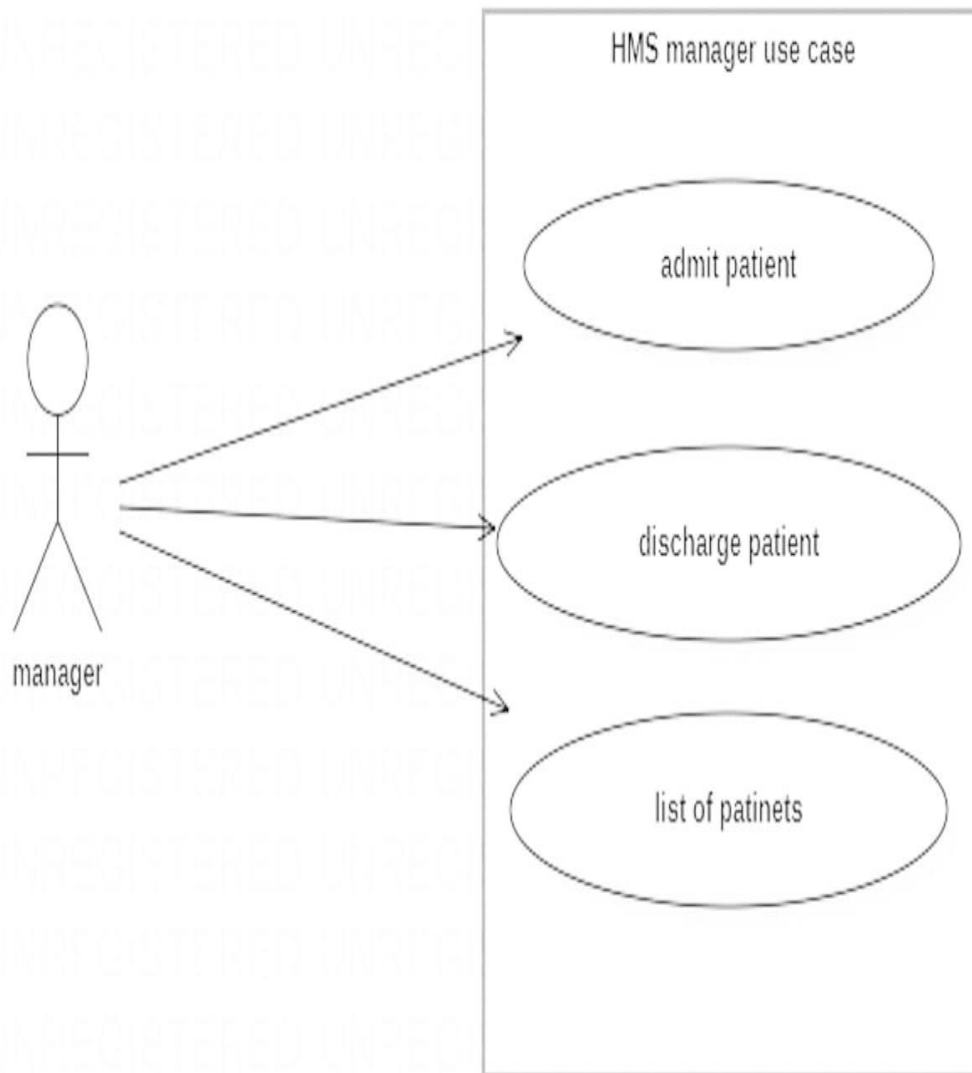


Use Case Model (Admin):



Use case diagram (admin)

Use Case Model (User):



Use case diagram (user)

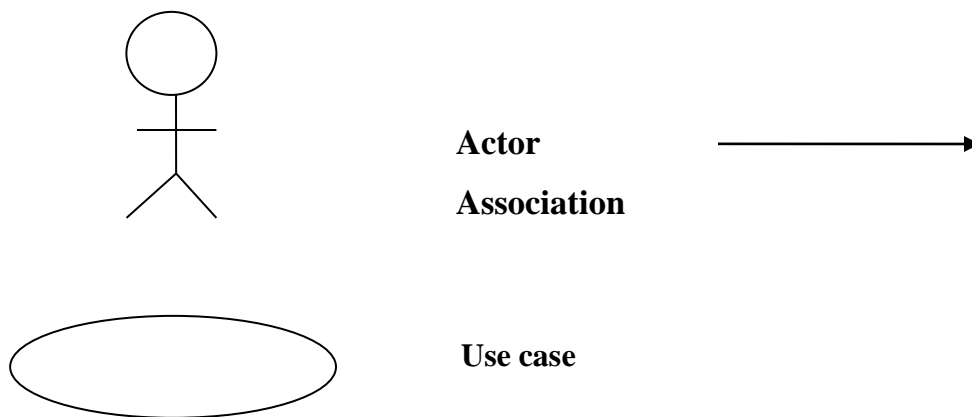
Use Case diagram description:

A use case diagram at its simplest is a representation of user interaction with the system and depicting the specifications of use case. A use case diagram can portray the different type of user of a system and the various ways that they interact with the system.

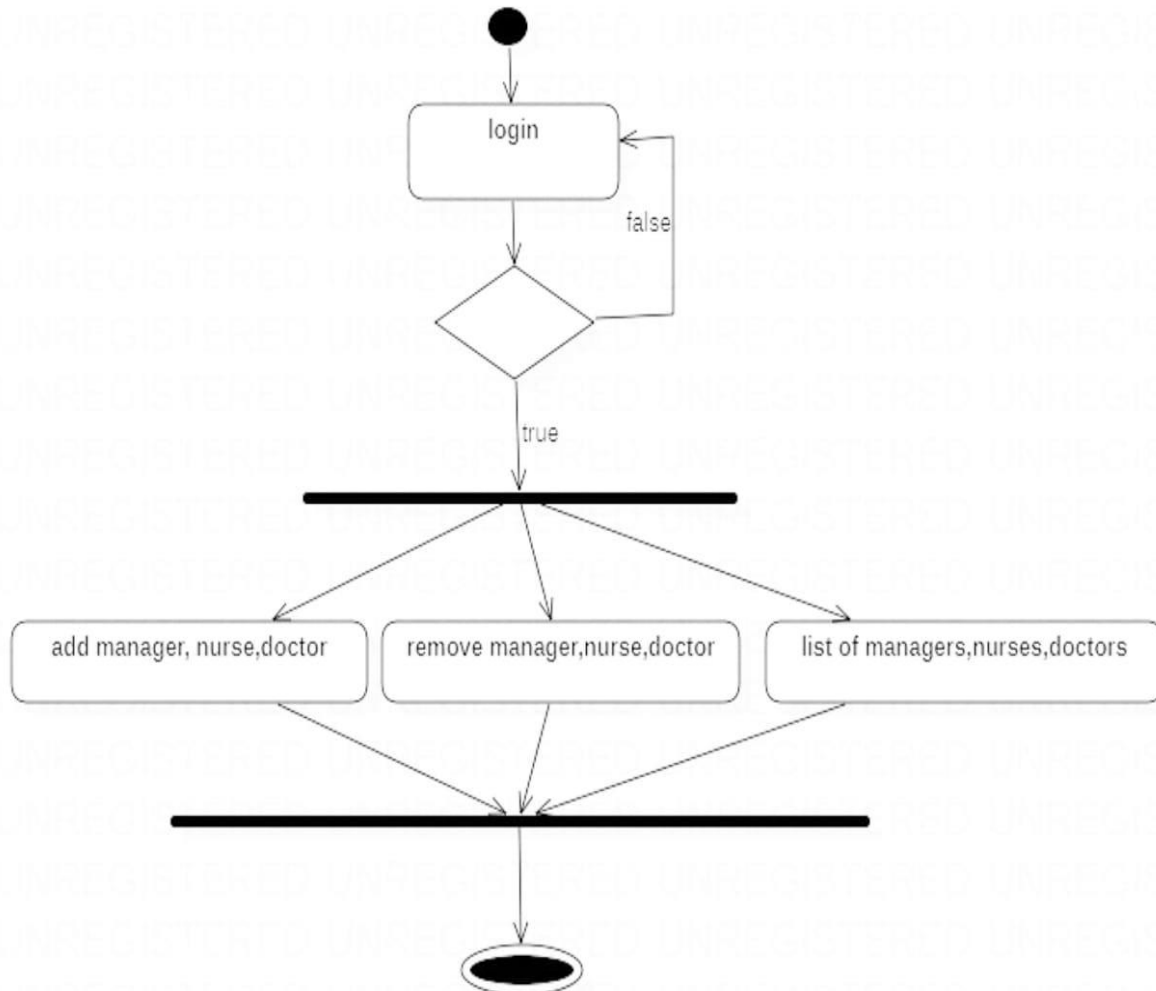
The type of diagram is typically used in conjunction with the textual use case and will often to be accomplished by other type of diagrams as well.

With regards to use case diagrams, that are exactly what they are meant to do, while a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that “Use case diagrams are the blueprints for the system”. They provide the simplified and graphical representation of what the system must actually do.

- Always structure and organize the use case diagram from the perspective of the actor.
- Use case should start off simple and at the highest view possible. Only they can be refined and detailed further.
- Use case diagrams are based upon functionality and thus should focus on the “WHAT” and not the “HOW”.

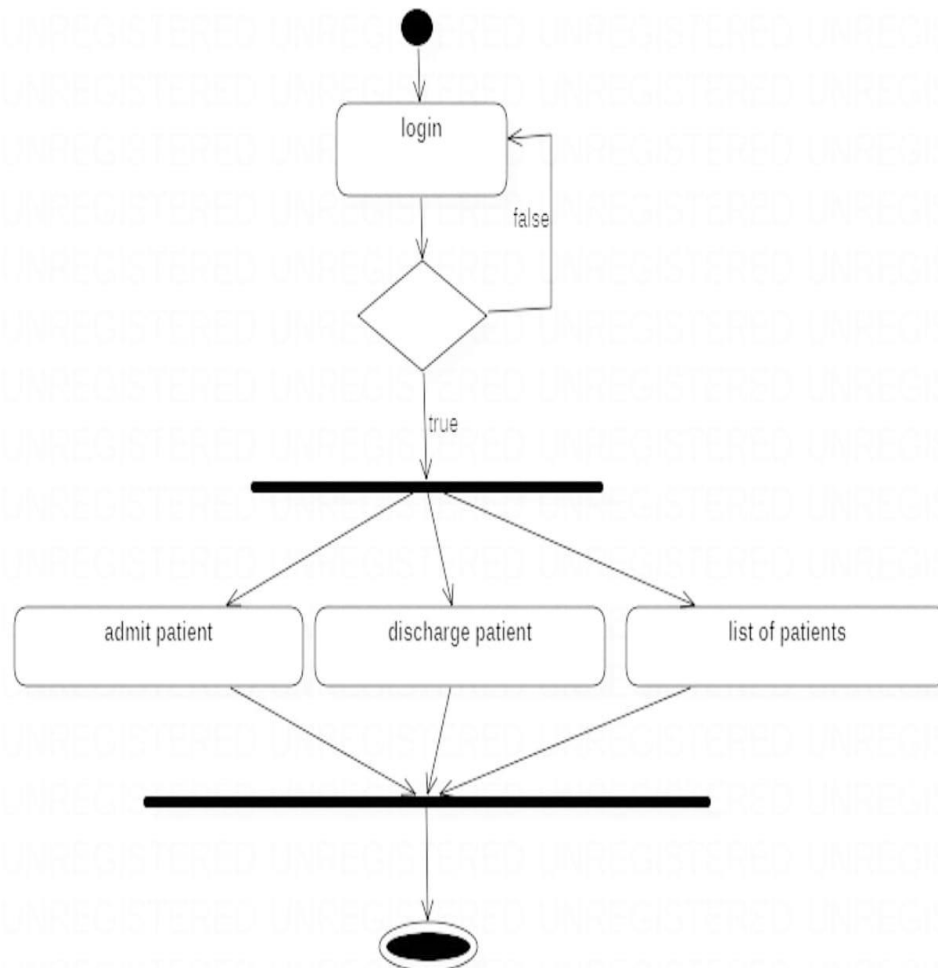


Activity Diagram (Admin):



Activity diagram (admin)

Activity Diagram (User):



Activity Diagram (User):

Activity diagram description:

Activity diagrams are graphical representations of workflows of stepwise activity and actions with support for choice, iteration and concurrency. In the unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow control. The most important shape types:

- Rounded rectangles represent actions.
- Diamonds represent decisions.
- Bars represent the start (split) or end (join) of concurrent activities.
- The black circle represents the start (initial state) of the workflow.
- An encircled black circle represents the end (final state).

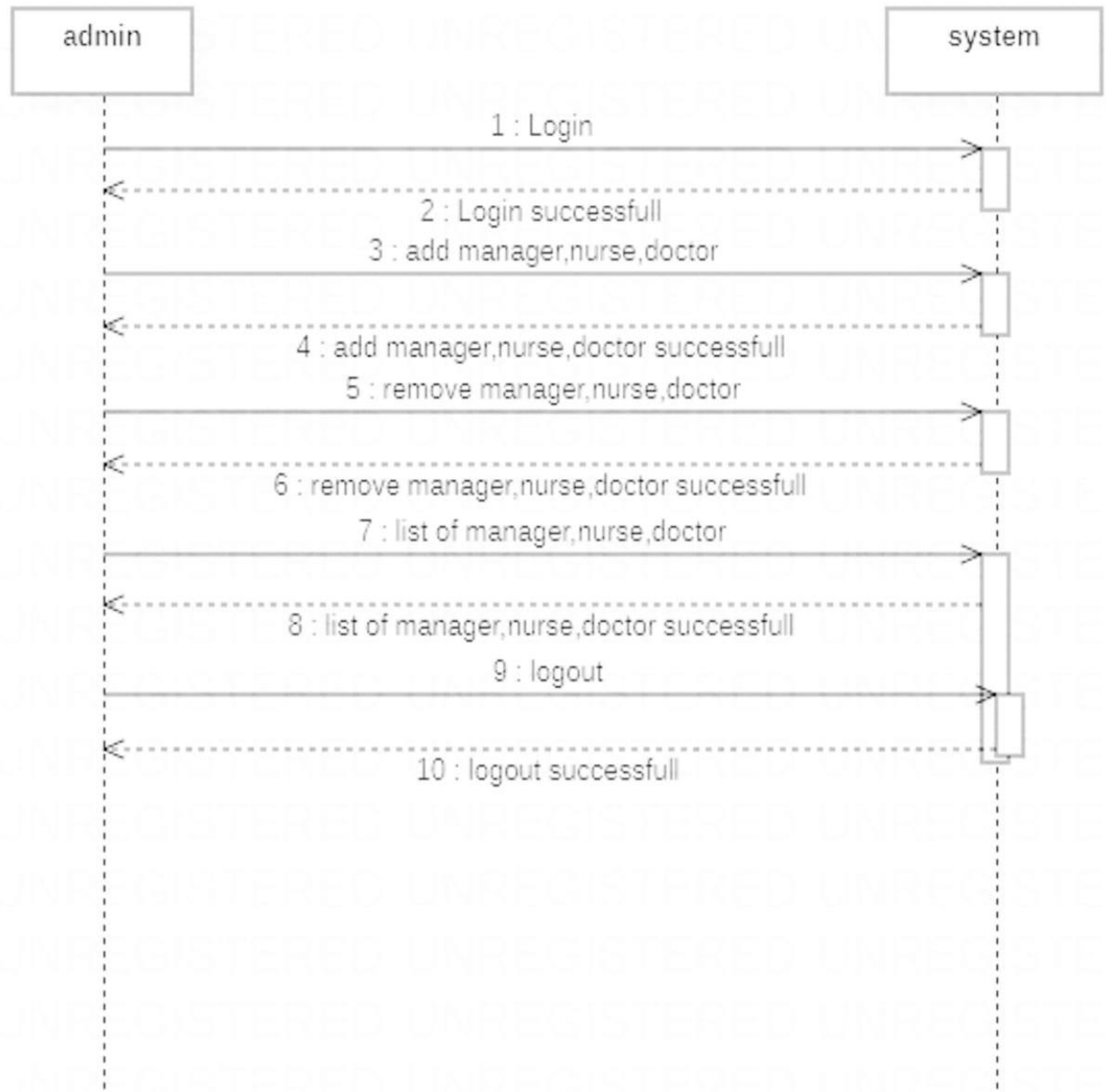
Arrows run from the start towards the end represent the order in which activities happen.

Activity diagrams may be regarded as a form of flow chart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

It was a specialized form of state diagrams, and it was renormalized to be based on Petri net-like semantics, increasing the scope of situations that can be modeled using activity diagrams.

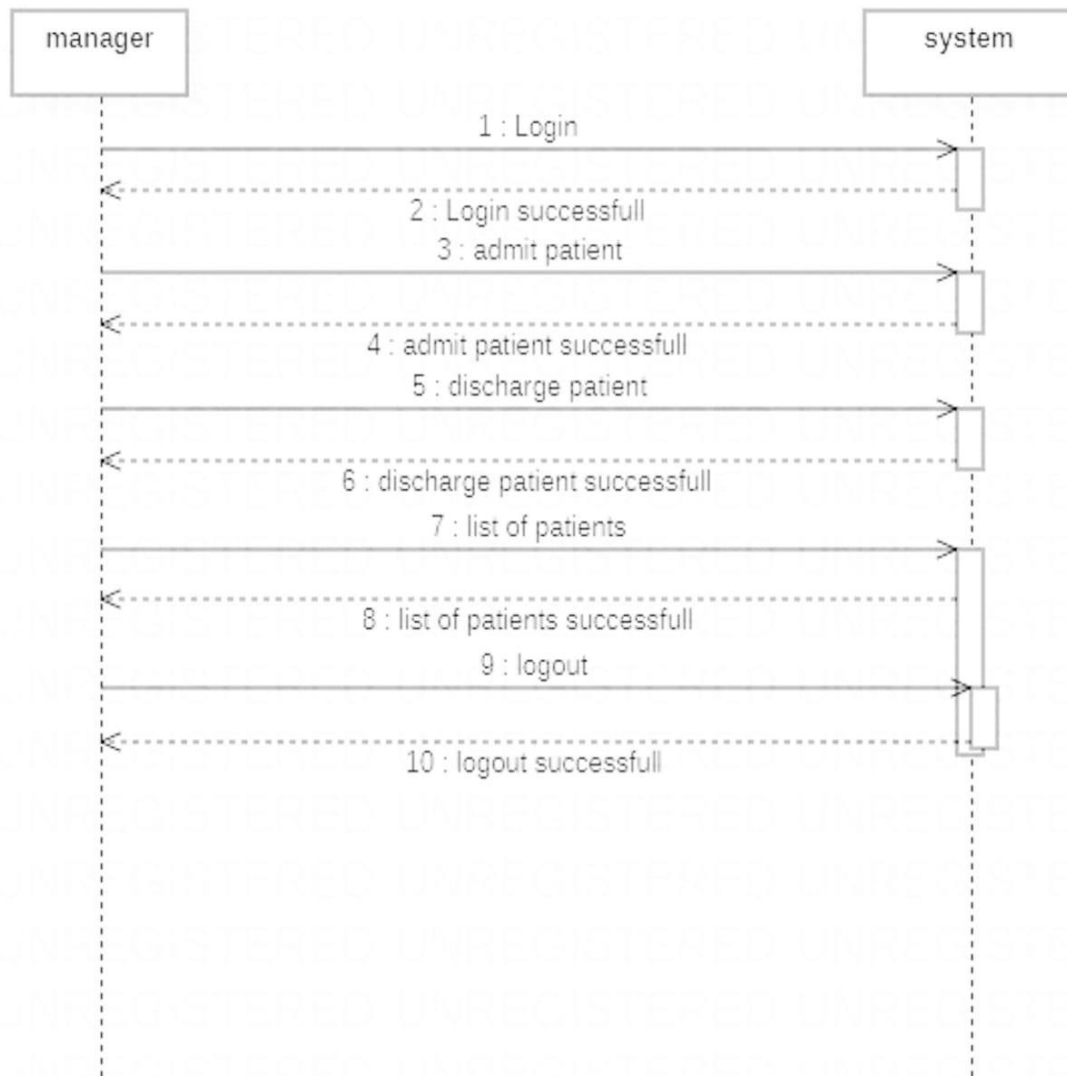
UML activity diagrams can be used in various domains, i.e. in design of embedded systems, it is possible to verify such a specification using model checking technique.

Sequence Diagram (Admin):



Sequence diagram (admin)

Sequence Diagram (User):



Sequence Diagram (User):

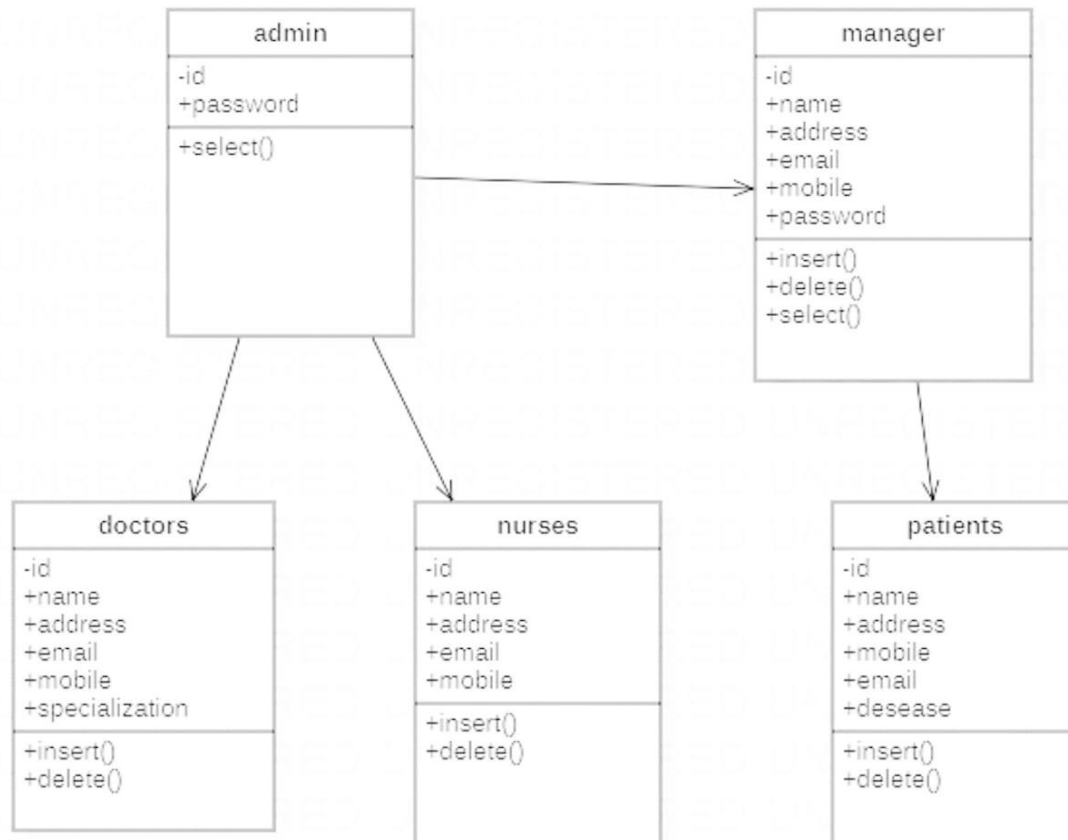
Sequence diagram description:

Sequence diagrams descriptions among classes. These interactions are modeled as exchange of messages. These diagram focus on classes and the messages they exchange to accomplish some described behavior. Sequence diagrams type of interaction diagrams.

- Class roles, which represent roles that objects may play within the interaction.
- Lifelines, which represent the existence of an object over a period of time.
- Activations, which represent the time during which an object is performing an operation.
- Messages, which represent communication between objects.

A sequence diagram shows the participants in an interaction and the sequence of message among them. A sequence diagram shows the interaction of a system with its actor.

Class Diagram:

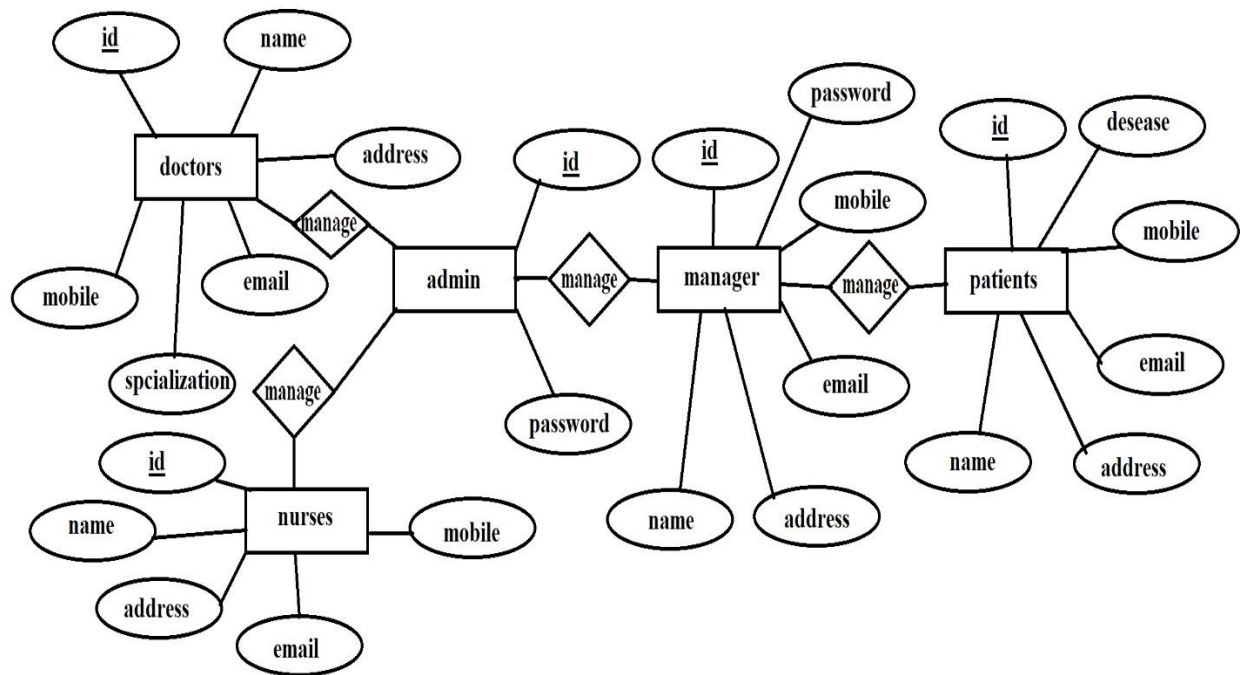


Class Diagram

Class diagram description:

In software engineering a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object orientated modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

ER Diagram



ER diagram

ER diagram description:

In 1976, chain developed the entity relationship (ER) model, a high-level data model that is useful in a developing in a conceptual design for a database, creation of an ER Diagram, which is one of the first steps in designing a database helps the designer to understand and to specify the desired components. An ER model is a diagram containing entities or “items”, relationships among them, and attributes of the entities and relationships. The entity relationship model a high level data model that is useful in developing a conceptual design for a database. The overall logical expressed graphical by an ER-diagram. It consists a set of basic objects entities and of relationship among these objects. An entity is represented by the set of attributes. Attributes are descriptive properties possessed by each member of an entity set. The ER diagram consists of the following major components.

The entity relationship diagram is graphical representation of the database logic and includes a detailed description of all entities, relationship and constraints. The ER diagram is based on perception of a real world that consists of a set of basic object called entities and of relationship among these objects. It is pictorial representation; it is easy to understand such a diagram of following main component-

- Rectangles represent entity set.
- Ellipses represent relationship set.
- Diamonds represent relationship set.
- Double ellipses represent multivalve attribute.

Home Page

```
<?php
?>

<html>

<head>

<style>

body{

background-image:url("bg1.jpg");

background-position:center;

background-size:cover;

}

</style>

</head>

<body>

<h1
           style="position:absolute;top:100px;left:350px;width:800px;font-
family:'algerian';color:black"><b/>welcome to hospital managment system</h1>

<a
           href="admin
           login.php"><input
           type="button"
style="position:absolute;top:250px;left:450px;font-family:'algerian';font-
size:20px;height:50px;width:450px;background-color:yellow"   value="LOGIN  AS  A
DIRECTOR"></a>

<a
           href="manager
           login.php"><input
           type="button"
style="position:absolute;top:350px;left:450px;font-family:'algerian';font-
size:20px;height:50px;width:450px;background-color:yellow"   value="LOGIN  AS  A
MANAGER"></a>

</body>

</html>
```

Admin Login.php


```

<html>

<head>

<style>

body{

background-image:url("bg2.jpg");

background-position:center;

background-size:cover;

}

</style>

</head>

<form method="POST">

<h1 style="position:absolute;top:100px;left:120px;font-family:'algerian';color:black">LOGIN FORM FOR ADMIN</h1>

<h1 style="position:absolute;top:190px;left:50;color:black;left:100px">USER ID</h1>

<input type="text" name="user_id" placeholder="enter your user id here" style="position:absolute;top:260px;left:100px;height:35px;width:400px;border-width:2px" required>

<h1 style="position:absolute;top:280px;color:black;left:100px">PASSWORD</h1>

<input type="password" name="pwd" placeholder="enter your password" style="border-width:2px;position:absolute;top:345px;left:100px;height:35px;width:400px">

<input type="submit" value="LOGIN" style="position:absolute;top:410px;left:100px;height:40px;width:400px;color:blue;background-color:white;font-size:20px;font-family:algerian" required>

</form>

<body>

<button onclick="document.location='home.php'" style="background-color:yellow;width:150px">back</button>

</body>

</html>

<?php

```

```

include('conn.php');
$id=$_POST['user_id'];
$password=$_POST['pwd'];
if(empty($password) and empty($id))
{
    echo"<h3 style='color:red;position:absolute;left:100px;top:450px'><u>". "PLESAE
ENTER USERID AND PASSWORD"."</u></h3>";
}
else
{
    $query="select * from admin where id='$id' and password='$password'";
    $q=mysqli_query($conn,$query);
    $num=mysqli_num_rows($q);
    if($num==1)
    {
        header('location:admin work.php');
    }
    else
    {
        echo"<h3 style='color:red;position:absolute;left:100px;top:450px'><u>". "PLEASE
ENTER CORRECT USERID AND PASSWORD"."</u></h3>";
    }
}
?>

```

Manager login.php

```

<html>
<head>
<style>

```

```

body{
background-image:url("bg2.jpg");
background-position:center;
background-size:cover;
}
</style>
</head>
<form method="POST">
<h1
style="position:absolute;top:100px;left:120px;font-
family:'algerian';color:black">LOGIN FORM FOR MANAGER</h1>
<h1 style="position:absolute;top:190px;left:50;color:black;left:100px">EMAIL ID</h1>
<input type="text" name="email" placeholder="enter your email
id"style="position:absolute;top:260px;left:100px;height:35px;width:400px;border-
width:2px" required>
<h1 style="position:absolute;top:280px;color:black;left:100px">PASSWORD</h1>
<input type="password" name="pwd" placeholder="enter your password"
style="border-
width:2px;position:absolute;top:345px;left:100px;height:35px;width:400px">
<input type="submit" value="LOGIN"
style="position:absolute;top:410px;left:100px;height:40px;width:400px;color:blue;backgr
ound-color:white;font-size:20px;font-family:algerian" required>
</form>
<body>
<button onclick="document.location='home.php'" style="background-
color:yellow;width:150px">back</button>

</body>
</html>
<?php
include("conn.php");
$email=$_POST['email'];

```

```

$password=$_POST['pwd'];
if(empty($password) and empty($email))
{
    echo"<h3 style='color:red;position:absolute;left:100px;top:450px'><u>". "PLESAE
ENTER EMAIL AND PASSWORD"."</u></h3>";
}
else
{
    $query="select * from manager where email='$email' and password='$password'";
    $q=mysqli_query($conn,$query);
    $num=mysqli_num_rows($q);
    if($num==1)
    {
        header('location:manager work.php');
    }
    else
    {
        echo"<h3 style='color:red;position:absolute;left:100px;top:450px'><u>". "PLEASE
ENTER CORRECT EMAIL AND PASSWORD"."</u></h3>";
    }
}
?>

Admin work.php

<?php
?>

<html>

<head>

<style>

```

```

body{
background-image:url("bg1.jpg");
background-position:center;
background-size:cover;
}
</style>
</head>
<body>

<button><h1
style="position:absolute;top:50px;left:300px;width:800px;font-
family:'algerian'"><b>welcome to HOSPITAL managment system what you want
do</h1></button>

<a
href="add
doctors.php"><input
type="button"
style="position:absolute;top:200px;left:50px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow"
value="ADD DOCTOR
FOR HOSPITAL"></a>

<a
href="remove
doctor.php"><input
type="button"
style="position:absolute;top:300px;left:50px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow"
value="REMOVE
DOCTOR FROM HOSPITAL"></a>

<a
href="doctors
list.php"><input
type="button"
style="position:absolute;top:400px;left:50px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow"
value="LIST OF
DOCTORS PRESENT IN HOSPITAL"></a>

<a
href="add
nurses.php"><input
type="button"
style="position:absolute;top:200px;left:500px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow"
value="ADD NURSE FOR
HOSPITAL"></a>

<a
href="remove
nurse.php"><input
type="button"
style="position:absolute;top:300px;left:500px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow"
value="REMOVE NURSE
FROM HOSPITAL"></a>

<a
href="nurses
list.php"><input
type="button"
style="position:absolute;top:400px;left:500px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow"
value="LIST OF NURSES
PRESENT IN HOSPITAL"></a>

```

```
<a href="add manager.php"><input type="button"
style="position:absolute;top:200px;left:950px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow" value="ADD MANAGER
FOR HOSPITAL"></a>
```

```
<a href="remove manager.php"><input type="button"
style="position:absolute;top:300px;left:950px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow" value="REMOVE
MANAGER FROM HOSPITAL"></a>
```

```
<a href="manager list.php"><input type="button"
style="position:absolute;top:400px;left:950px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow" value="LIST OF
MANAGERS PRESENT IN HOSPITAL"></a>
```

```
<button onclick="document.location='home.php'" style="background-
color:yellow;width:150px;height:40px">logout</button>
```

```
</body>
```

```
</html>
```

Manager work.php

```
<?php
```

```
?>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body{
```

```
background-image:url("bg1.jpg");
```

```
background-position:center;
```

```
background-size:cover;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<button><h1 style="position:absolute;top:50px;left:300px;width:800px;font-family:'algerian'"><b/>welcome to HOSPITAL managment system what you want to do</h1></button>
```

```
<a href="admit patients.php"><input type="button" style="position:absolute;top:200px;left:500px;font-family:'algerian';font-size:20px;height:50px;width:400px;background-color:yellow" value="ADMIT PATIENT IN HOSPITAL"></a>
```

```
<a href="descharge.php"><input type="button" style="position:absolute;top:300px;left:500px;font-family:'algerian';font-size:20px;height:50px;width:400px;background-color:yellow" value="DISCHARGE PATIENT FROM HOSPITAL"></a>
```

```
<a href="patients list.php"><input type="button" style="position:absolute;top:400px;left:500px;font-family:'algerian';font-size:20px;height:50px;width:400px;background-color:yellow" value="LIST OF PATIENTS PRESENT IN HOSPITAL"></a>
```

```
<button onclick="document.location='home.php'" style="background-color:yellow;width:150px;height:45px">logout</button>
```

```
</body>
```

```
</html>
```

Add or admin .php

```
<html>
```

```
<head>
```

```
<style>
```

```
body{
```

```
background-image:url("bg2.jpg");
```

```
background-position:center;
```

```
background-size:cover;
```

```
}
```

```
</style>
```

```
</head>
```

```
<form method="POST">
```

```
<h1 style="position:absolute;top:120px;left:50px;color:black;font-family:'algerian'">ADMIT PATIENT IN HOSPITAL</h1>
```

```
<input type="text" name="id" placeholder="GIVE ID TO PATIENT"
style="position:absolute;top:360;left:350px;height:35px;width:215px" required>
```

```
<input type="text" name="name" placeholder="ENTER NAME"
style="position:absolute;top:220px;left:50px;height:35px;width:215px" required>
```

```
<input type="text" name="address" placeholder="ENTER ADDRESS"
style="position:absolute;top:220px;left:350px;height:35px;width:215px" required>
```

```
<input type="text" name="mobile" placeholder="ENTER MOBILE NUMBER"
style="position:absolute;top:290;left:50px;height:35px;width:215px" required>
```

```
<input type="text" name="email" placeholder="ENTER EMAIL ID"
style="position:absolute;top:290;left:350px;height:35px;width:215px" required>
```

```
<input type="text" name="disease" placeholder="NAME OF DISEASE"
style="position:absolute;top:360;left:50px;height:35px;width:215px" required>
```

```
<input type="submit" value="DONE"
style="position:absolute;top:430;left:50px;height:35px;width:520px;color:white;background-color:blue;font-size:20px;font-family:'algerian'">
```

```
</form>
```

```
<body>
```

```
<button onclick="document.location='manager work.php'" style="background-color:yellow;width:150px">back</button>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
include("conn.php");
```

```
$na=$_POST['name'];
```

```
$address=$_POST['address'];
```

```
$mobile=$_POST['mobile'];
```

```
$email=$_POST['email'];
```

```
$dis=$_POST['disease'];
```

```
$id=$_POST['id'];
```



```
if(empty($na) or empty($mobile) or empty($address) or empty($dis) or empty($email) or empty($id))
```

```
{
```

```
    echo'<h3                style="position:absolute;top:460px;left:30px;color:red;font-family:algerian"><u><blink>'. "please enter data required data with correct manner". "</blink></h3>";
```

```
}
```

```
else
```

```
{
```

```
$query1="select * from patients where email=$email;";
```

```
$q1=mysqli_query($conn,$query1);
```

```
$num=mysqli_num_rows($q1);
```

```
    if($num>=1)
```

```
    {
```

```
        echo'<h3                style="position:absolute;top:460px;left:50px;color:red;font-family:algerian"><u><blink>'. "the patient is already in your hospital". "</blink></h3>";
```

```
    }
```

```
    else
```

```
    {
```

```
        $query="insert into patients values('$id','$na','$address',$mobile,$email,$dis);";
```

```
$q=mysqli_query($conn,$query);
```

```
    if($q)
```

```
    {
```

```
        echo'<h3                style="position:absolute;top:460px;left:100px;color:red;font-family:algerian"><u><blink>'. "patient admitted". "</blink></h3>";
```

```
    }
```

```
    }  
}  
?>
```

Remove or discharge .php

```
<html>  
<head>  
<style>  
body{  
background-image:url("bg2.jpg");  
background-position:center;  
background-size:cover;  
}  
</style>  
</head>  
<form method="POST">  
<h1 style="position:absolute;top:100px;left:100px;font-family:'algerian';color:black">discharge patient FROM HOSPITAL</h1>  
<h1 style="position:absolute;top:190px;left:100px;color:black;left:100px">ENTER EMAIL OF PATIENT</h1>  
<input type="text" name="email" placeholder="enter email here" style="position:absolute;top:260px;left:100px;height:35px;width:400px;border-width:2px" required>  
<h1 style="position:absolute;top:280px;color:black;left:100px">ENTER DISEASE OF PATIENT</h1>  
<input type="text" name="dis" placeholder="enter disease" style="border-width:2px;position:absolute;top:345px;left:100px;height:35px;width:400px" required>  
<input type="submit" value="remove" style="position:absolute;top:410px;left:100px;height:40px;width:400px;color:blue;background-color:white;font-size:20px;font-family:algerian" required>  
</form>
```

```
<button    onclick='document.location='manager    work.php''    style='background-color:yellow;width:150px'>back</button>
```

```
<body>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
include("conn.php");
```

```
$dis=$_POST['dis'];
```

```
$email=$_POST['email'];
```

```
if(empty($pwd) and empty($email))
```

```
{
```

```
    echo"<h3  style='color:red;position:absolute;left:100px;top:450px'><u>". "PLESAE  
ENTER DISEASE AND EMAIL"."</u></h3>";
```

```
}
```

```
else
```

```
{
```

```
$query="select * from patients where disease='$dis' and email='$email'";
```

```
$q=mysqli_query($conn,$query);
```

```
$num=mysqli_num_rows($q);
```

```
if($num==1)
```

```
{
```

```
    $q1="delete from patients where disease='$dis' and email='$email'";
```

```
    $que=mysqli_query($conn,$q1);
```

```
    if($que)
```

```
    {
```

```
        echo"<h3  
style='color:red;position:absolute;left:100px;top:450px'><u>". "PATIENT  DISCHARGE  
SUCCESSFULLY"."</u></h3>";
```

```

    }}
else
{
    echo"<h3      style='color:red;position:absolute;left:100;top:450px'><u>"."THIS
PATIENT IS NOT IN HOSPITAL"."</u></h3>";
}}
?>

```

List .php

```

<html>
<head>
<style>
body{
background-image:url("bg1.jpg");
background-position:center;
background-size:cover;
}
table{
    position:absolute;
    top:150;
    left:350;
    background-color:YELLOW;
}
</style>
</head>
<body>
<h1      style="position:absolute;top:20;left:400px;font-
family:algerian;color:red;width:380px"><u>here  is  the  list  of  all  managers
</u></h1></body></html>

```

```
<button    onclick='document.location='admin    work.php'"    style='background-  
color:yellow;width:150px">back</button>
```

```
<?php
```

```
include("conn.php");
```

```
$q1="select * from manager;";
```

```
    $query1=mysqli_query($conn,$q1);
```

```
    echo"<center/><table border='1'>
```

```
    <tr>
```

```
    <th>ID</th>
```

```
    <th>NAME</th>
```

```
    <th>ADDRESS</th>
```

```
    <th>MOBILE</th>
```

```
    <th>EMAIL</th>
```

```
    <th>PASSWORD</th>
```

```
    </tr>";
```

```
    while($row=mysqli_fetch_array($query1))
```

```
{
```

```
    echo "<tr><td>".$row["id"]."</td> <br/><td>";
```

```
    echo $row["name"]." </td><br/><td>";
```

```
    echo $row["address"]."</td><td>";
```

```
    echo $row["mobile"]." </td><br/><td>";
```

```
    echo $row["email"]." </td><br/><td>";
```

```
    echo $row["password"]." </td><br/><td>";
```

```
    echo "</tr>";
```

```
}
```

```
echo"</table>";
```

```
?>
```

5.1 Implementation:

System implementation is the process of making the newly designed systems fully operational. The system is implemented after careful testing. The primary goal of product implementation is development of source code that is easy to read and easy to understand. The term implementation has different meaning, ranging from the conversion of a basic application to a compatible replacement of a computer system. Implementation is used here to mean the process converting a new or a revised system design in to an operational one. During the implementation stage we convert the detailed code in a programming language. Clarity of source code eases debugging, testing and modification of a software product. the difficulties encountered during implementation is successful integration of source code components into a functioning system before a routine can be placed in the evolving system. It may be required that the routine be inspected by an inspection team. Or reviewed or tested to a given level of test coverage.

Testing

Software testing is an important element of s/w quality assurance and represents the ultimate review of specification, design and coding. the increasing visibility of s/w as a system element and the costs associated with an s/w failure are motivating forces for well planned, through testing thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing. testing is a set of activities that can be planned in advance and conducted systematically. testing is very important stages of a software include unit testing, integration testing and deployment testing.

White box testing:

Is a test case design philosophy that uses the control structure described as part of component-level design to derive test cases.

White box testing is also called glass-box testing or structural testing. It performs close examination of procedural details. They test the software test the software logical path by having test cases exercising specific sets of condition and loops. It examine the program status at various points to determine whether the expected status correspond to the actual one.

Internal working of the product test can be conducted to ensure that internal operational performs according to specification. And all component have been adequately exercised is called white-box testing.

Using white box testing method it can be ensured that:

- All independent parts within a module have been exercised at least one.
- Exercise all logical decision and their all false sides.
- Exercise all loop at their boundaries and within their operational bound.
- Exercise internal data structure to ensure that validity.

Black box testing:

It is also called functional testing or behavioral testing. It performs tests of software interfaces. They test the operation ability of software functional, acceptance of input and delivery of output and the maintainability of external information integrity.

It is testing without knowledge of the internal working of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the “legal” inputs and what the expected outputs should be, but now the program actually arrives at those outputs. It is because of this that black box testing can be considered testing with respect to the specifications, no other knowledge of program in necessary.

It examine some of the basic aspects of a system, having little regard for the internal logical software structure.

Black box testing is not an alternative to white box technique. It is complementary approach that is likely to uncover a different class than white box methods.

- Black box testing attempts to find error in the categories.
- Incorrect or missing function.
- Interface error.
- Errors in data structure or database access.
- Performance errors.
- Initialization and termination errors.

10. Reports & Table Figure

Admin Table

```
C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe
5 rows in set (0.00 sec)

mysql> desc admin;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id     | varchar(20) | YES | | NULL | |
| password | varchar(20) | YES | | NULL | |
+-----+
2 rows in set (0.08 sec)

mysql>
```

manager Table, Doctor table and Nurses table

```
C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe

mysql> desc manager;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id     | varchar(20) | NO | PRI | NULL | |
| name   | varchar(20) | YES | | NULL | |
| address | varchar(30) | YES | | NULL | |
| mobile | int(11) | YES | | NULL | |
| email  | varchar(20) | YES | | NULL | |
| password | varchar(20) | YES | | NULL | |
+-----+
6 rows in set (0.08 sec)

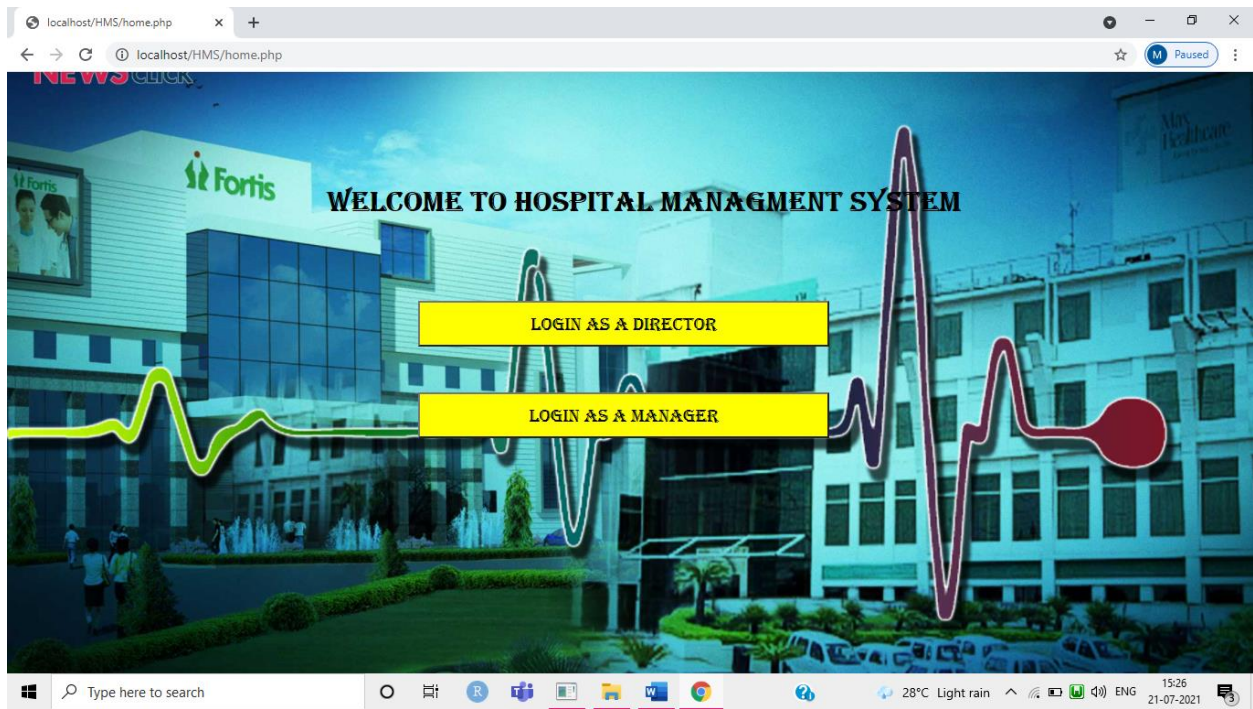
mysql> desc doctors;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id     | varchar(20) | NO | PRI | NULL | |
| name   | varchar(20) | YES | | NULL | |
| address | varchar(30) | YES | | NULL | |
| mobile | int(11) | YES | | NULL | |
| email  | varchar(20) | YES | | NULL | |
| specialization | varchar(20) | YES | | NULL | |
+-----+
6 rows in set (0.09 sec)

mysql> desc nurses;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id     | varchar(20) | NO | PRI | NULL | |
| name   | varchar(20) | YES | | NULL | |
| address | varchar(30) | YES | | NULL | |
| mobile | int(11) | YES | | NULL | |
| email  | varchar(20) | YES | | NULL | |
+-----+
5 rows in set (0.07 sec)

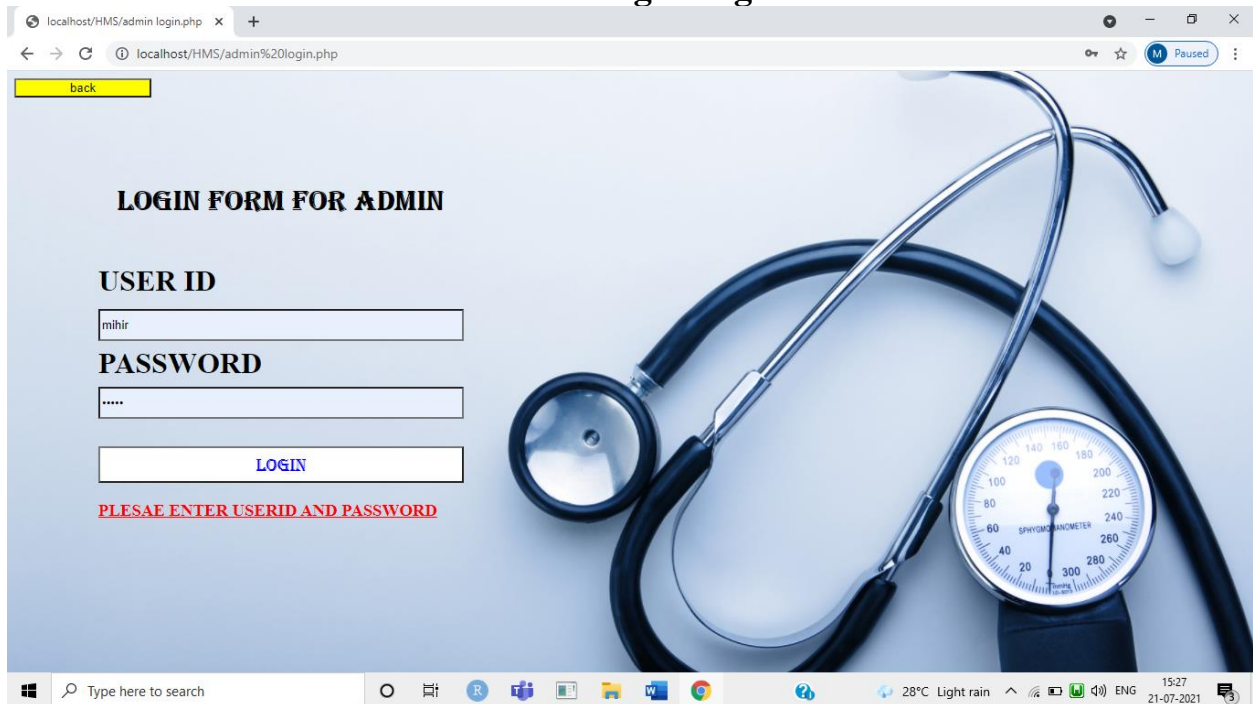
mysql>
```

11. Screenshots of Project

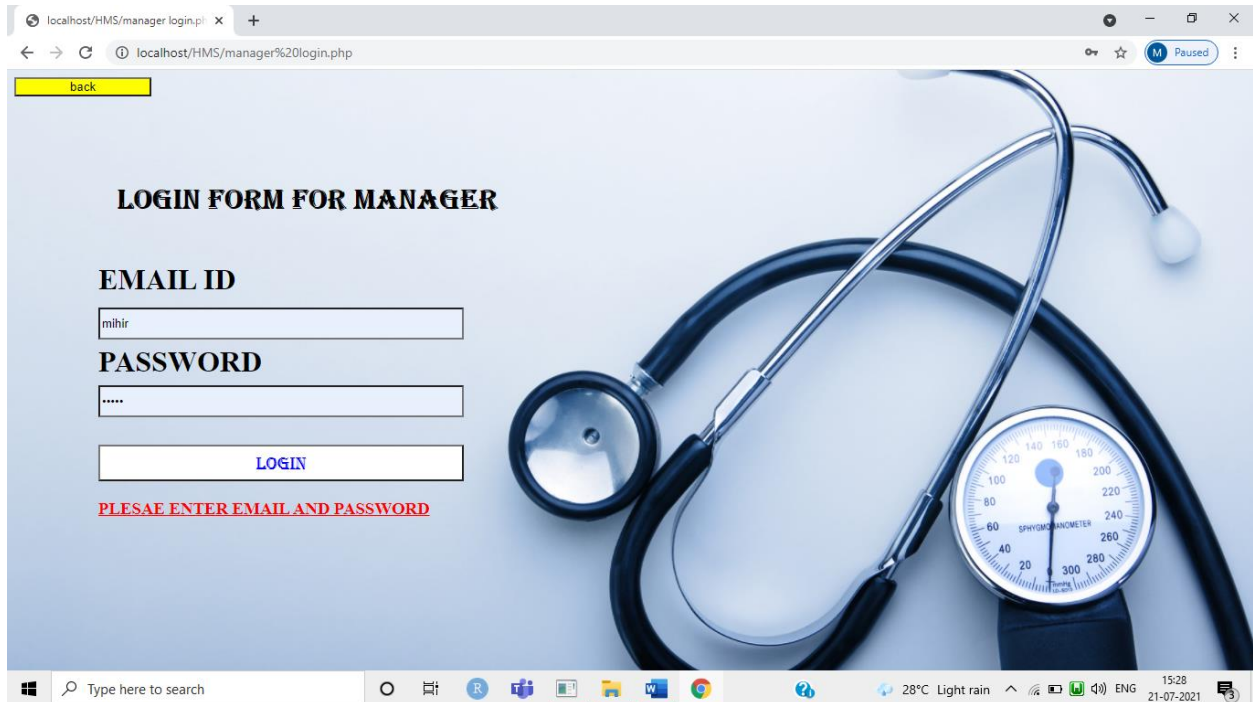
Home Page Page



admin login Page



manager login page



The screenshot shows a web browser window with the address bar displaying 'localhost/HMS/manager login.php'. The page has a light blue background with a large image of a stethoscope and a blood pressure gauge on the right side. The main heading is 'LOGIN FORM FOR MANAGER'. Below it, there are two input fields: 'EMAIL ID' with the value 'mihir' and 'PASSWORD' with the value '*****'. A 'LOGIN' button is positioned below the password field. A red error message 'PLESAE ENTER EMAIL AND PASSWORD' is displayed at the bottom of the form area. The browser's taskbar at the bottom shows the Windows search bar, several application icons, and system information including '28°C Light rain' and the date '21-07-2021'.

back

LOGIN FORM FOR MANAGER

EMAIL ID

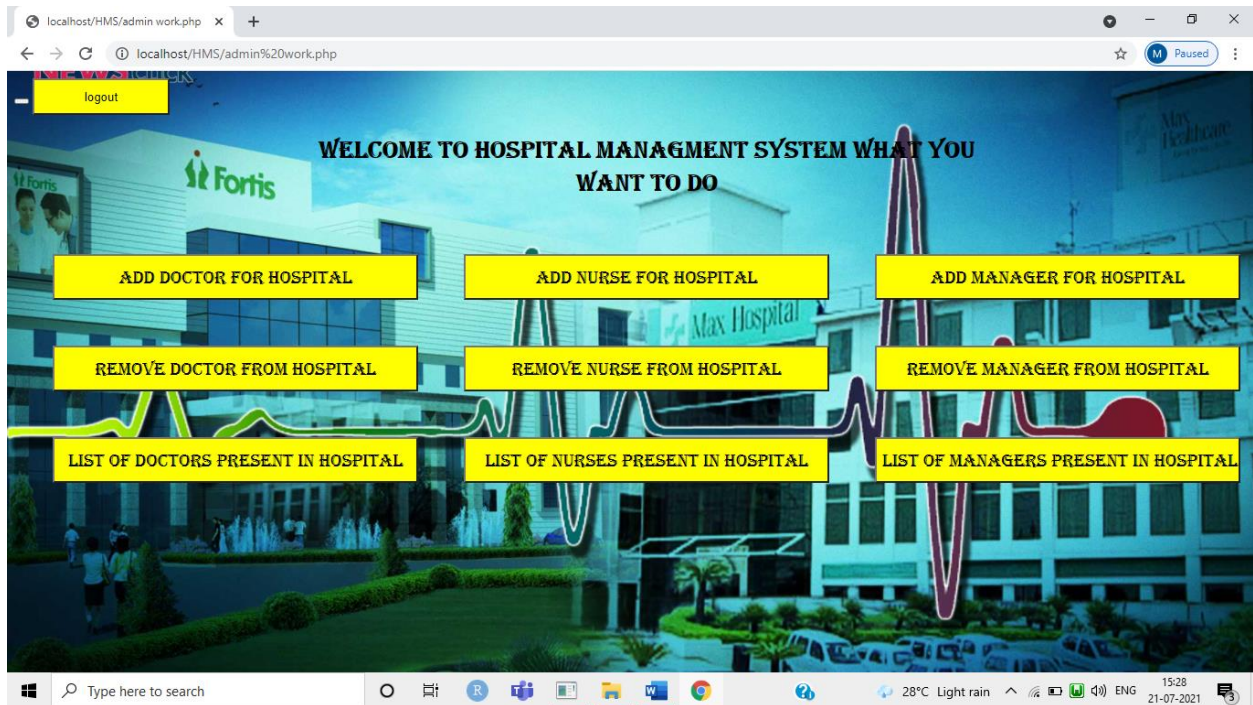
mihir

PASSWORD

LOGIN

PLESAE ENTER EMAIL AND PASSWORD

admin work list page



The screenshot shows a web browser window with the address bar displaying 'localhost/HMS/admin work.php'. The page features a background image of a modern hospital building with 'Fortis' and 'Max Hospital' signage. A large red heart rate line graphic is overlaid on the image. The main heading is 'WELCOME TO HOSPITAL MANAGMENT SYSTEM WHAT YOU WANT TO DO'. Below the heading, there are nine yellow buttons arranged in a 3x3 grid. The buttons are: 'ADD DOCTOR FOR HOSPITAL', 'ADD NURSE FOR HOSPITAL', 'ADD MANAGER FOR HOSPITAL', 'REMOVE DOCTOR FROM HOSPITAL', 'REMOVE NURSE FROM HOSPITAL', 'REMOVE MANAGER FROM HOSPITAL', 'LIST OF DOCTORS PRESENT IN HOSPITAL', 'LIST OF NURSES PRESENT IN HOSPITAL', and 'LIST OF MANAGERS PRESENT IN HOSPITAL'. A 'logout' button is located in the top left corner. The browser's taskbar at the bottom shows the Windows search bar, several application icons, and system information including '28°C Light rain' and the date '21-07-2021'.

logout

WELCOME TO HOSPITAL MANAGMENT SYSTEM WHAT YOU WANT TO DO

ADD DOCTOR FOR HOSPITAL

ADD NURSE FOR HOSPITAL

ADD MANAGER FOR HOSPITAL

REMOVE DOCTOR FROM HOSPITAL

REMOVE NURSE FROM HOSPITAL

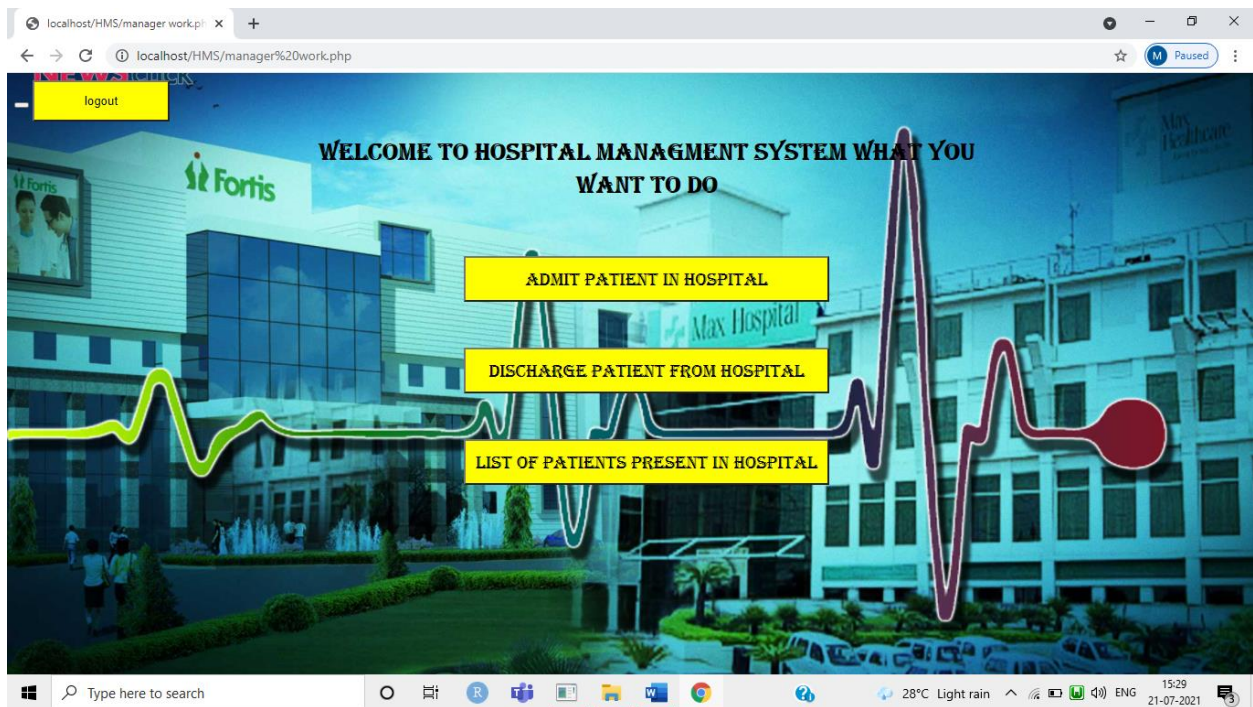
REMOVE MANAGER FROM HOSPITAL

LIST OF DOCTORS PRESENT IN HOSPITAL

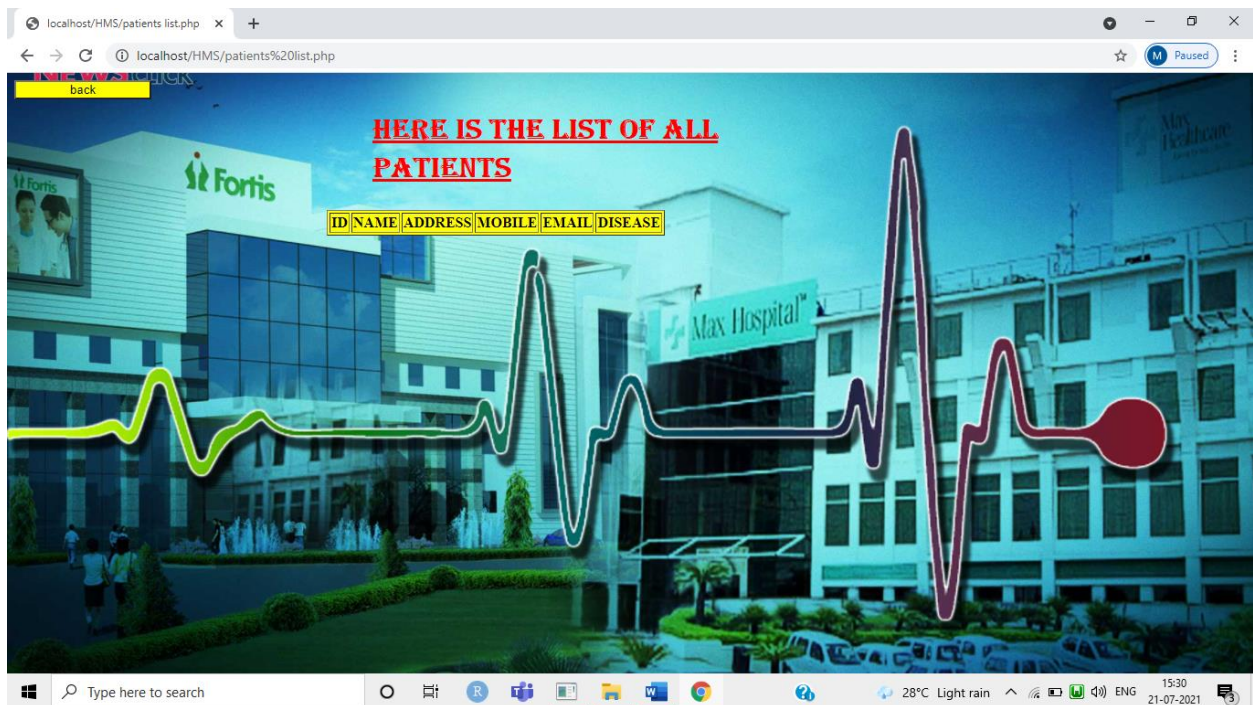
LIST OF NURSES PRESENT IN HOSPITAL

LIST OF MANAGERS PRESENT IN HOSPITAL

Manger work page



list page



Add or admit page

localhost/HMS/admit patients.php

back

ADMIT PATIENT IN HOSPITAL

ENTER NAME

ENTER ADDRESS

ENTER MOBILE NUMBER

ENTER EMAIL ID

NAME OF DISEASE

GIVE ID TO PATIENT

DONE

PLEASE ENTER DATA REQUIRED DATA WITH CORRECT MANNER

Type here to search

28°C Light rain

15:30
21-07-2021

Remove or discharge page

localhost/HMS/discharge.php

back

DISCHARGE PATIENT FROM HOSPITAL

ENTER EMAIL OF PATIENT

enter email here

ENTER DISEASE OF PATIENT

enter disease

REMOVE

PLESAE ENTER DISEASE AND EMAIL

Type here to search

28°C Light rain

15:31
21-07-2021

12. Conclusion

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

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

13. Future Scope & Further Enhancement of Project

Main feature of the existing system this HMS (hospital management system) can hold limited number of information and data regarding patients, doctors, and nurses. As mentioned earlier this system is a manual system which manages all this information in the paper based book known a register. Following are some changes we can do in these system in future.

- Online payment will available.
- Data delete & edit system will available for all section.
- User account will verify by Mobile SMS will available in this system.
- Diagnostics billing system.

14. References

- www.w3school.com
- www.3pdf.com
- www.project.com
- www.ukessay.com
- www.elarninglarning.com