**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya Indore**



**Project Report on**

**Hospital Management System**

**Guided by:-**

Dr. Shaligram Prajapat sir

**Submitted by :-**

Nitin Dwivedi

IT-2K19-37

**BONAFIDE CERTIFICATE**

Certified that this project titled " Hospital Management System " is a bonafide work of **Nitin Dwivedi (IT-2K19-37)** who carried out the research and completed the project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form part of any other project on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Internal examiner External examiner

**ACKNOWLEDGEMENT**

It is indeed with a great pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Director **Dr. B.K.Tripathi**, International Institute of Professional Studies, for the facilities provided to accomplish this main project.

We feel elated in manifesting our sense of gratitude tour internal project guide **Dr. Shaligram Prajapat, Associate Professor,** International Institute of Professional Studies. He has been a constant source of inspiration for us and we are very deeply thankful to him for his support and valuable advice.

We express our heartful thanks to all of our friends who helped us in successful completion of this project.

**ABSTRACT**

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

**Table of Contents**

[1. INTRODUCTION 7](#_Toc88337413)

[**1.1 INTRODUCTION** 7](#_Toc88337414)

[**1.2 AIM** 7](#_Toc88337415)

[**1.3 Objective** 7](#_Toc88337416)

[**1.4 SCOPE** 8](#_Toc88337417)

[**1.5 EXISTING SYSTEM** 8](#_Toc88337418)

[**1.6 PROPOSED SYSTEM** 9](#_Toc88337419)

[2. FEASIBILITY STUDY 9](#_Toc88337420)

[**2.1 OPERATIONAL FEASIBILITY** 9](#_Toc88337421)

[**2.2 TECHNICAL FEASIBILITY** 10](#_Toc88337422)

[**2.3 ECONOMICAL FEASIBILITY** 10](#_Toc88337423)

[3. SOFTWARE REQUIREMENTS SPECIFICATION 11](#_Toc88337424)

[**3.1 Hardware Requirements** 11](#_Toc88337425)

[**3.2 Software Requirements** 11](#_Toc88337426)

[4. DESIGN & PLANNING 12](#_Toc88337427)

[**4.1 Software Development Life Cycle Model** 12](#_Toc88337428)

[**4.1.1 WATERFALL MODEL** 12](#_Toc88337429)

[**4.2 Use case diagram:** 13](#_Toc88337430)

[**4.3 Class Diagram:** 13](#_Toc88337431)

[**4.4 Sequence diagram:** 14](#_Toc88337432)

[**4.5 Activity diagram:** 15](#_Toc88337433)

[**4.6 Deployement diagram:** 15](#_Toc88337434)

[**4.7 E-R Diagram:** 16](#_Toc88337435)

[**4.8 Data Flow Diagram:** 18](#_Toc88337436)

[**4.8.1 0 Level DFD** 18](#_Toc88337437)

[**4.8.2 1st Level DFD** 19](#_Toc88337438)

[**4.8.3 2nd Level DFD** 20](#_Toc88337439)

[**4.8.4 3rd Level DFD** 21](#_Toc88337440)

[5. IMPLEMENTATION: 22](#_Toc88337441)

[**5.1 Introduction:** 22](#_Toc88337442)

[**5.2 Sample code:** 22](#_Toc88337443)

[6. TESTING 36](#_Toc88337444)

[**6.1 INTRODUCTION TO SYSTEM TESTING:** 36](#_Toc88337445)

[7. ADVANTAGES 41](#_Toc88337446)

[8. CONCLUSION 41](#_Toc88337447)

[9. REFERENCES 41](#_Toc88337448)

# 1. INTRODUCTION

## **1.1 INTRODUCTION**

Our project hospital management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id. The hospital management system can be entered using a username and password. It is accessible either by an administrator or receptionist. 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.

## **1.2 AIM**

This project is aimed to automate the hospital management system. This project is developed mainly to administrate doctor's appointment with the patients. The purpose of the project entitled as HOSPITAL MANAGEMENT SYSTEM is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully. The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details; while system output is to get these details on to the CRT screen.

## **1.3 Objective**

1) Define hospital

2) Recording information about the Patients that come.

3) Generating bills.

4) Recording information related to diagnosis given to Patients.

5) Keeping record of the Immunization provided to children/patients.

6) Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff andDoctors. All these works are done on papers.

## **1.4 SCOPE**

Our project hospital management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id. The hospital management system can be entered using a username and password. It is accessible either by an administrator or receptionist. 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.

## **1.5 EXISTING SYSTEM**

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread through out the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

## **1.6 PROPOSED SYSTEM**

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

# 2. FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1) Operational Feasibility

2) Technical Feasibility

3) Economical Feasibility

## **2.1 OPERATIONAL FEASIBILITY**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

## **2.2 TECHNICAL FEASIBILITY**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

## **2.3 ECONOMICAL FEASIBILITY**

Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need of online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

# 3. SOFTWARE REQUIREMENTS SPECIFICATION

## **3.1 Hardware Requirements**

|  |  |
| --- | --- |
| **Number** | **Description** |
| 1 | PC with 250 GB or more Hard disk. |
| 2 | PC with 2 GB RAM. |
| 3 | PC with Pentium 1 and Above. |

## **3.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Type** |
| 1 | Operating System | Windows XP / Windows |
| 2 | Language | PHP |
| 3 | Database | MySQL |
| 4 | IDE | Visual Code |
| 5 | Browser | Google Chrome |

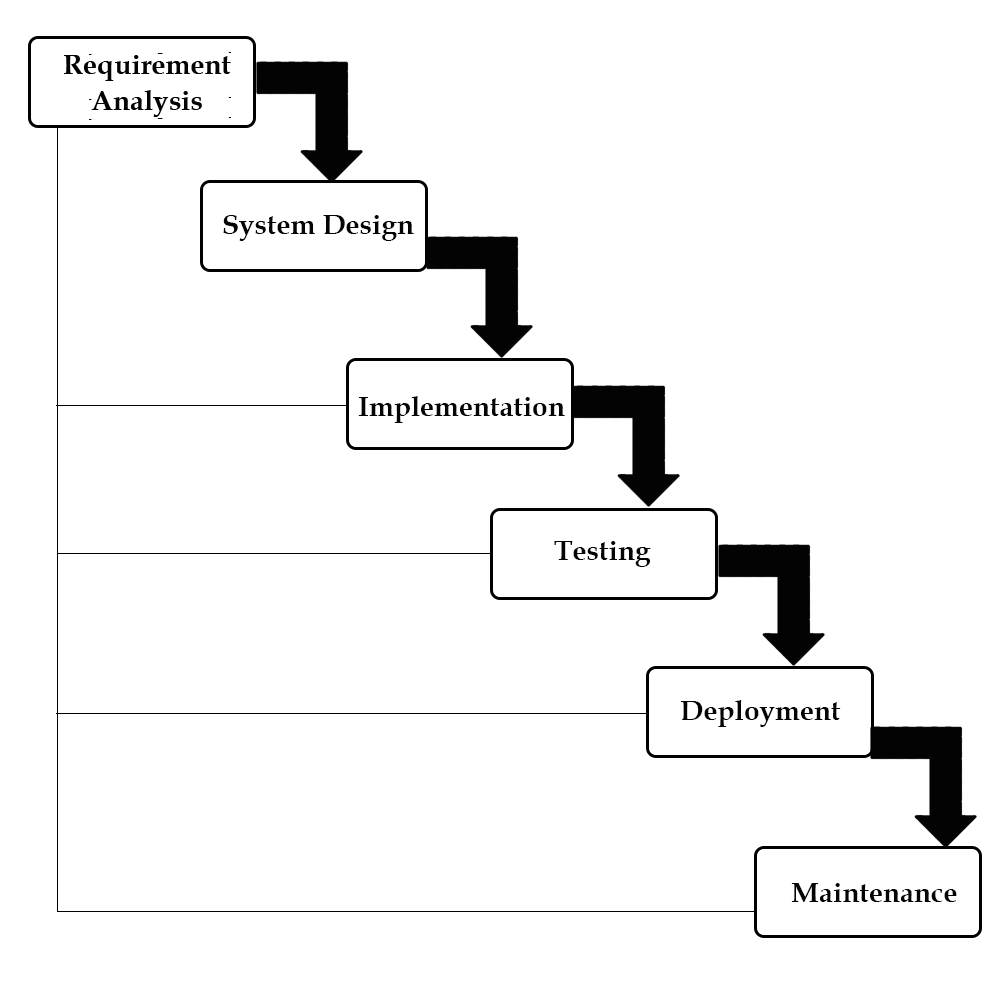
# 4. DESIGN & PLANNING

## **4.1 Software Development Life Cycle Model**

### **4.1.1 WATERFALL MODEL**

The waterfall model was selected as the SDLC model due to the following reasons:

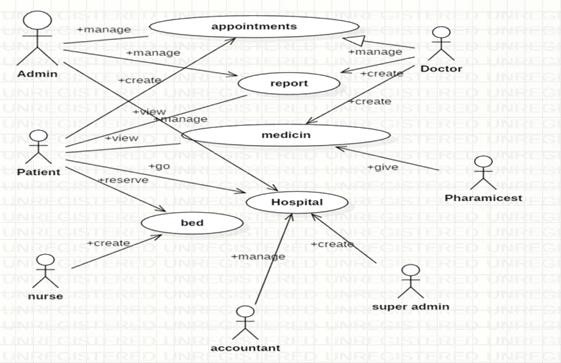
* Requirements were very well documented, clear and fixed.
* Technology was adequately understood.
* Simple and easy to understand and use.
* There were no ambiguous requirements.
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Clearly defined stages.
* Well understood milestones. Easy to arrange tasks.



## **4.2 Use case diagram:**

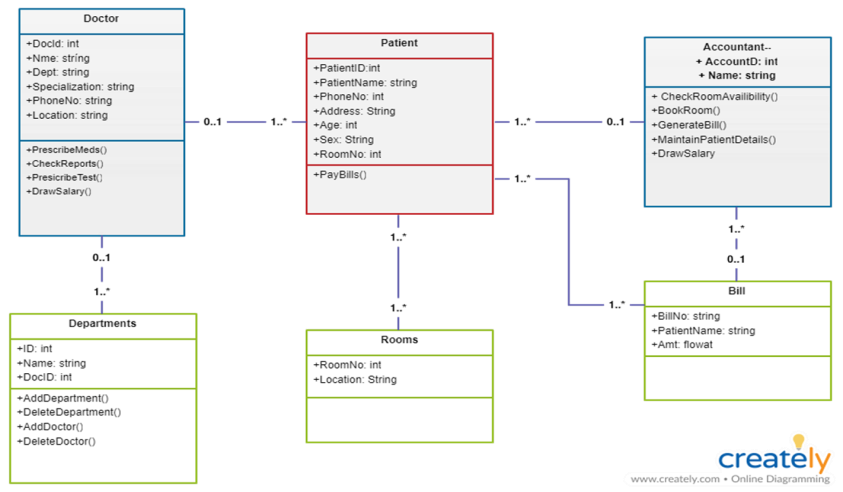
A use case diagram in the Unified Modelling Language(UML) is a type of behavioural diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases.

Use case diagrams are formally included in two modelling languages defined by the OMG:theunfied modeling language(UML) and the systems modelling language(sysML)



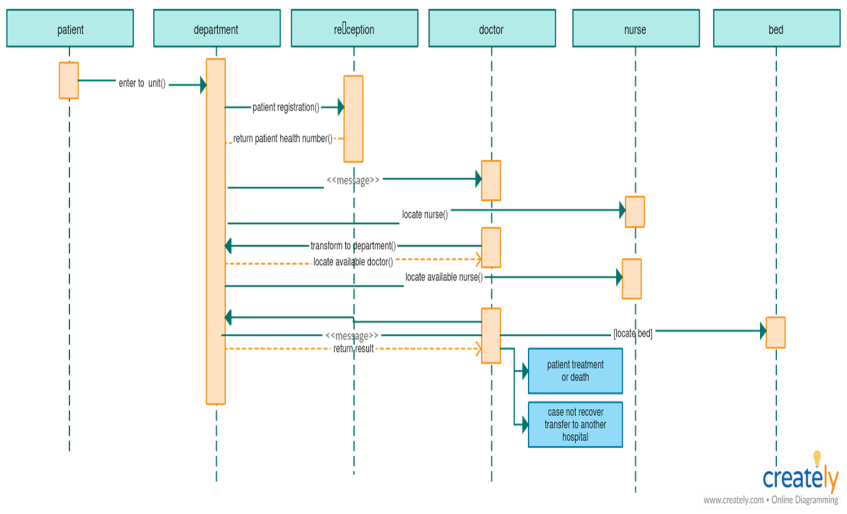
## **4.3 Class Diagram:**

A Class is a category or group of things that has similar attributes and common behaviour. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provides the representation that developers work from. Class diagrams help on the analysis side, too.



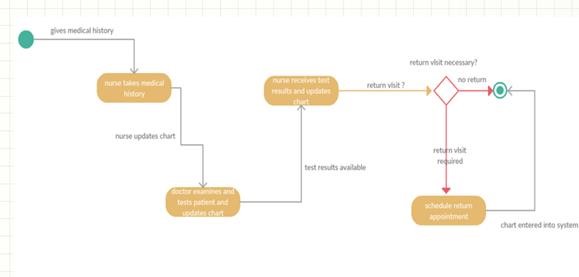
## **4.4 Sequence diagram:**

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



## **4.5 Activity diagram:**

An **Activity diagram** is another important behavioural diagram in [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity.



## **4.6 Deployement diagram:**

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

**Hospital Local Server**

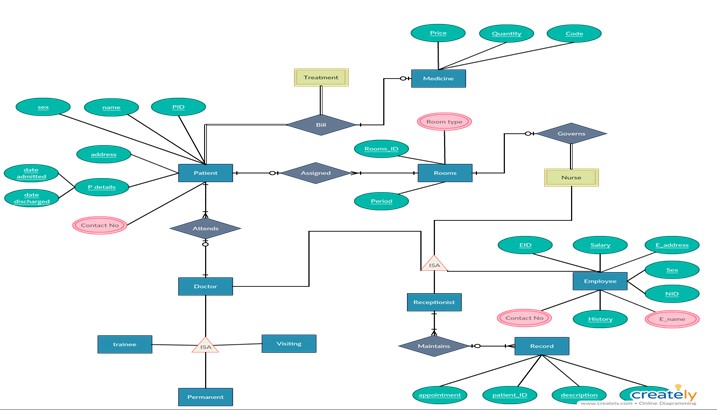
**desktop Client**

**Printer**

**Database Server**

## **4.7 E-R Diagram:**

Database is absolutely an integral part of software system. To fully utilize ER Diagram in database engineering guarantee you to produce high quality database design to use in database creation, management and maintenance. An ER model also provides a means for communication.



## **4.8 Data Flow Diagram:**

### **4.8.1 0 Level DFD**

Hospital Management System



4. Data Flow Diagrams




        DFD: Level 0




            16
 

### **4.8.2 1st Level DFD**

Hospital Management System




      DFD: Level 1




           17
 

### **4.8.3 2nd Level DFD**

Hospital Management System




       DFD: Level 2




           18
 

### **4.8.4 3rd Level DFD**

Hospital Management System




       DFD: Level 3




           19
 

# 5. IMPLEMENTATION:

## **5.1 Introduction:**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

## **5.2 Sample code:**

**Home.html:**

<!DOCTYPE html>

<html>

<body>

<table width="1350" height="640" border="1" >

<tr>

<td colspan="2" style="background-color:#FFF5EE;">

<h1>HOSPITAL MANAGEMENT SYSTEM</h1>

<h3 align="center">ADMIN PANEL</h3>

</td>

</tr>

<tr>

<td style="background-color:#00FFFF;width:50px;height:400px;">

<table align="center">

<tr><td><form action="doctor.php" align="center">

<input type="submit" align="center" value=" doctor ">

</form></td>

</tr>

<tr>

<td><form action="nurse.php" align="center">

<input type="submit" align="center" value=" nurse ">

</form></td>

</tr>

<tr>

<td><form action="patient.php" align="center">

<input type="submit" align="center" value=" patient ">

</form></td>

</tr>

<tr>

<td><form action="pharmacist.php" align="center">

<input type="submit" align="center" value=" pharamacist ">

</form></td>

</tr>

<tr>

<td><form action="laboratorist.php" align="center">

<input type="submit" align="center" value=" laboratorist ">

</form></td><tr>

<td><form action="accountant.php" align="center">

<input type="submit" align="center" value=" accountant ">

</form></td>

</tr>

</table>

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;height:400px;"><h3 align="center">Advanced, powerfull, flexible complete management software for hospital, clinic and medical institutes. Integrates and facilitates all user area of a hospital: </h3><h4>align="center">Administrator</h4>

<h4 align="center">Doctor</h4>

<h4 align="center">Patient</h4>

<h4 align="center">Nurse</h4>

<h4 align="center">Pharmacist</h4>

<h4 align="center">Laboratorist</h4>

<h4 align="center">Accountant</h4>

</td>

</tr>

<tr>

<td colspan="2" style="background-color:#9ACD32;text-align:center;">

<table align="right">

<th>

<tr>

<form action="appointment.php" align="center">

<input type="submit" align="center" value=" appointment ">

</form>

</tr>

<tr>

<form action="payment.php" align="center">

<input type="submit" align="center" value=" payment ">

</form>

</tr>

<tr>

<form action="bloodbank.php" align="center">

<input type="submit" align="center" value=" bloodbank ">

</form>

</tr>

<tr>

<form action="medicine.php" align="center">

<input type="submit" align="center" value=" medicine ">

</form>

</tr>

<tr>

<form action="operations.php" align="center">

<input type="submit" align="center" value=" operations ">

</form>

</tr>

<tr>

<form action="birthreport.php" align="center">

<input type="submit" align="center" value=" birthreport ">

</form>

</tr>

<tr>

<form action="deathreport.php" align="center">

<input type="submit" align="center" value=" deathreport ">

</form>

</tr>

<tr>

<form action="bedallotment.php" align="center">

<input type="submit" align="center" value=" bedallotment ">

</form>

</tr>

</th>

</table>

</td>

</tr></table></body></html>

**Doctor.PHP**

<!DOCTYPE html>

<html>

<body>

<table width="1350" height="640" border="1" >

<tr>

<td colspan="2" style="background-color:#FFF5EE;">

<h1>HOSPITAL MANAGEMENT SYSTEM</h1>

<h3 align="center">ADMIN PANEL</h3>

</td>

</tr>

<tr>

<td style="background-color:#00FFFF;width:50px;height:400px;">

<table align="center">

<tr>

<td><form action="nurse.php" align="center">

<input type="submit" align="center" value=" nurse ">

</form></td>

</tr>

<tr>

<td><form action="patient.php" align="center">

<input type="submit" align="center" value=" patient ">

</form></td>

</tr>

<tr>

<td><form action="pharmacist.php" align="center">

<input type="submit" align="center" value=" pharamacist ">

</form></td>

</tr>

<tr>

<td><form action="laboratorist.php" align="center">

<input type="submit" align="center" value=" laboratorist ">

</form></td>

<tr>

<td><form action="accountant.php" align="center">

<input type="submit" align="center" value=" accountant ">

</form></td>

</tr>

</table>

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;height:400px;">

<?php

$host='localhost';

$username='root';

$password='';

$dbname='hospital';

$con=mysql\_connect($host,$username,$password);

mysql\_select\_db($dbname);

$result = mysql\_query("SELECT \* FROM doctor");

echo "<h4 align='center'> doctors list </h4>";

echo "<table border=1 align=center><tr><th>s.no</th><th>name</th><th>d\_id</th><th>qualification</th><th>speciality</th><th>age</th></tr>";

while($row = mysql\_fetch\_array($result))

{

echo "<tr>";

echo "<td>" . $row['s\_no'] . "</td>";

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

echo "<td>" . $row['d\_id'] . "</td>";

echo "<td>" . $row['qualification'] . "</td>";

echo "<td>" . $row['speciality'] . "</td>";

echo "<td>" . $row['age'] . "</td>";

echo "</tr>";

}

echo "</table>";

mysql\_close($con);

?>

<br><br>

<table align="right">

<th>

<tr>

<form action="adddoctor.php" align="center">

<input type="submit" align="center" value=" add new doctor ">

</form>

</tr>

<tr>

<form action="deletedoctor.php" align="center">

<input type="submit" align="center" value=" delete doctor ">

</form></tr>

<tr>

<form action="viewcompletedoctor.php" align="center">

<input type="submit" align="center" value=" viewcomplete ">

</form>

</tr>

<tr>

<form action="admin.html" align="center">

<input type="submit" align="center" value=" home ">

</form>

</tr></table>

</td>

</tr><tr>

<td colspan="2" style="background-color:#9ACD32;text-align:center;">

<table align="right">

<th>

<tr><form action="appointment.php" align="center">

<input type="submit" align="center" value=" appointment ">

</form>

</tr><tr><form action="payment.php" align="center">

<input type="submit" align="center" value=" payment ">

</form>

</tr><tr><form action="bloodbank.php" align="center">

<input type="submit" align="center" value=" bloodbank ">

</form>

</tr><tr>

<form action="medicine.php" align="center">

<input type="submit" align="center" value=" medicine ">

</form>

</tr><tr><form action="operations.php" align="center">

<input type="submit" align="center" value=" operations ">

</form>

</tr><tr>

<form action="birthreport.php" align="center">

<input type="submit" align="center" value=" birthreport ">

</form>

</tr><tr><form action="deathreport.php" align="center">

<input type="submit" align="center" value=" deathreport ">

</form>

</tr><tr><form action="bedallotment.php" align="center">

<input type="submit" align="center" value=" bedallotment ">

</form>

</tr></th> </table>

</td></tr></table>

</body>

</html>

**Appointment.php**

<!DOCTYPE html>

<html>

<body>

<table width="1350" height="640" border="1" ><tr>

<td colspan="2" style="background-color:#FFF5EE;">

<h1>HOSPITAL MANAGEMENT SYSTEM</h1>

<h3 align="center">DOCTOR PANEL</h3>

</td>

</tr>

<tr>

<td style="background-color:#00FFFF;width:50px;height:400px;">

<table align="center">

<tr> <td><form action="docappointment.php" align="center">

<input type="submit" align="center" value=" Appointment ">

</form> </td></tr>

<tr> <td><form action="docperscription.php" align="center">

<input type="submit" align="center" value=" perscription ">

</form> </td> </tr>

<tr> <td> <form action="docoperation.php" align="center">

<input type="submit" align="center" value=" Operation ">

</form> </td></tr>

<tr> <td><form action="docmedicines.php.php" align="center">

<input type="submit" align="center" value=" Add Medicines ">

</form></td></tr>

<tr> <td> <form action="doctests.php" align="center">

<input type="submit" align="center" value=" Add Tests ">

</form></td>

</table>

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;height:400px;">

<h2 align="center"> Appointments </h2>

<?php

$host='localhost';

$username='root';

$password='';

$dbname='hospital';

$con=mysql\_connect($host,$username,$password);

mysql\_select\_db($dbname);

$result = mysql\_query("SELECT \* FROM appointment WHERE d\_id='$a'");

echo "<table border=1 align=center> <tr> <th>s.no</th> <th>pid</th> <th>name</th> <th>problem</th> <th>date</th> <th>time</th> <th>status</th> <th> update</th> </tr>";

while($row = mysql\_fetch\_array($result))

{

echo "<tr>";

echo "<td>" . $row['s\_no'] . "</td>";

echo "<td>" . $row['p\_id'] . "</td>";

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

echo "<td>" . $row['problem'] . "</td>";

echo "<td>" . $row['date\_of\_app'] . "</td>";

echo "<td>" . $row['time\_of\_app'] . "</td>";

echo "<td>" . $row['status'] . "</td>";

echo "<td>" ;?> <form action="updateappointment.php" align="center" method="POST">

<input type="hidden" name="sno" value=" <?php echo $row['s\_no']; ?> ">

<input type="hidden" name="pid" value=' <?php echo $row['p\_id']; ?> '>

<input type="submit" align="center" value=" update ">

</form> <?php echo "<td>";

echo "</tr>";

}

echo "</table>";

mysql\_close($con);

?>

<br><br>

<table align="center">

<tr>

<td><form action="allappointment.php" align="center">

<input type="submit" align="center" value=" all Appointment ">

</form> </td>

<td><form action="pendingappointment.php" align="center">

<input type="submit" align="center" value=" pending Appointment ">

</form> </td>

<td> <form action="upcomingappointment.php" align="center">

<input type="submit" align="center" value=" upcoming appointment ">

</form> </td>

<td><form action="completedappointment.php" align="center">

<input type="submit" align="center" value=" completed Appointment ">

</form></td></table>

</td></tr>

<tr>

<td colspan="2" style="background-color:#9ACD32;text-align:center;">

<table align="center"> <tr> <td> Doctor name </td> <td> </td> <td> Doctor id </td> <td> </td> </tr> </table>

</td></tr>

</table></body></html>

# 6. TESTING

## **6.1 INTRODUCTION TO SYSTEM TESTING:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTING**:

**Unit testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**Integration Testing:**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Acceptance Testing:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

# 7. ADVANTAGES

1. Easy Access to Patient Data.
2. Cost Effective.
3. Better revenue management.
4. Increased Data Security & Retrieve-ability.

# 8. CONCLUSION

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

# 9. REFERENCES

1. <https://itsourcecode.com/sdm_downloads/hospital-management-system-php-project-with-source-code/>
2. <https://drive.google.com/file/d/0B1m77YndfAXlbmN4ZTYya2VTUGlDQnB6V3IwVWFrUQ/view?resourcekey=0-EGDIDcuqK>
3. <https://sites.google.com/site/shaligramiipsdavvindore/Core-Activities/it-511-is/assignment-21/rubrics>