A Project Report

on

HOSPITAL MANAGEMENT SYSTEM

Submitted in partial fulfilment of the
Requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of

Mr. Padmanabhan P.

Associate Professor

Department of Computer Science and Engineering

Submitted by

NASIR FIRDAUSH: 18SCSE1120026

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA
DECEMBER – 2021

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"HOSPITAL MANAGEMNET SYSTEM"** in partial fulfillment of the requirements for the award of the Bachelor of technology submitted in the School of Computer Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September-2021 to November -2021, under the supervision of **Mr. Padmanabhan P. Associate Professor** Department of Computer Science and Engineering, Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me for the award of any other degree of this or any other places.

Name: **Nasir Firdaush**
Admission No: **18SCSE1120026**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor
(Padmanabhan P. Associate Professor)

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **Name: Nasir Firdaush**, **Admission No: 18SCSE1120026** has been held on __________________ and his work is recommended for the award of **Bachelor of Technology Computer Science and Engineering.**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

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 Dean **Dr. Munish Sabharwal**, Galgotias University, for the facilities provided to accomplish this main project.

We feel elated in manifesting our sense of gratitude tour internal project guide **Mr. Padmanabhan P, Associate Professor**, Department of Computer Science and Engineering, Galgotias University. 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 extremely grateful to our Departmental staff members, Lab technicians and Non-teaching staff members for their extreme help throughout our project.

Finally we express our heartfelt thanks to all of our friends who helped us in successful completion of this project.

Project Associates:

Nasir Firdaush (18SCSE1120026)

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 retrieves 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.

KEYWORDS: MIS (Management Information System), .NET Framework, SQL Server, SQL Client, ASP.NET

TABLE OF CONTENT

CONTENT

List of Figures

CHAPTER-1
Introduction

1.1 Introduction:

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The 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.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

Introduction

1.2 Problem Introduction:

Lack of immediate retrievals: -

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time.

Lack of immediate information storage: -

The information generated by various transactions takes time and efforts to be stored at right place.

Lack of prompt updating: -

Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

Error prone manual calculation: -

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient's bill based on various treatments.

Preparation of accurate and prompt reports: -

This becomes a difficult task as information is difficult to collect from various register.

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 and Doctors. All these works are done on papers.

1.2.1 Tool and Technology Used:-

- **HARDWARE REQUIREMENTS:**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

- **HARDWARE REQUIREMENTS FOR PRESENT PROJECT**:

PROCESSOR	:	Intel dual Core, i3 or above
RAM	:	2 GB or
HARD DISK	:	80 GB

Introduction

➢ SOFTWARE REQUIREMENTS:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

- **SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:**

OPERATING SYSTEM	:	Windows 7/ XP/8
FRONT END	:	Html, css, java script.
SERVER SIDE SCRIPT	:	Php
DATABASE	:	Mysql

CHAPTER-2
Literature Survey

One of the major challenges existing hospital management systems face is around operational efficiency and wait times between different processes, departments and persons. This paper highlights such limitations of existing systems and proposes a RFID (Radio Frequency ID) and wireless sensor based, location and information management framework that facilitates real time tracking of hospital assets, personnel and patients as they move through pre-set procedures as part of daily activities of the hospitals. The system covers the visual simulation and providing ability to analyze the ongoing operations so they can be corrected to achieve increased process efficiency and service levels.

This paper reviews the HIS (Hospital Information Systems) which are widely used in many hospitals in China mainly to provide easier and faster way for daily medical tasks /activities with a GUI And provides for overcoming some of the limitations of HIS, e.g. HIS aims at improving quality of health care services but do not have way of evaluating /measuring those. So this paper proposes HSMS (Hospital Services Management System) which aims at improving quality of services, identifying cost reduction areas, analyses and evaluate /rate heath care services . The ability to evaluate the services facilitates hospital achieve higher Customer satisfaction scores and get a competitive edge against those hospitals which scoreless or use HIS and do not have ways of promoting the quality of their services.

Many enterprise projects get scrapped due to high costs involved in initial planning requirement gathering and design phase. The costs in this phase become unmanageable due to lot of unknown factors. Like lack of Subject area expertise, lack of knowledge on different Hospital enterprise functions

1) Patient admission

2) Patient Treatment planning

3) Order Entry

4) Execution of diagnostic and treatment procedures

5) Administrative documentation

6) Billing

7) Clinical documentation

8) Discharge and

9) Referral to specialized medical institutions, lack of knowledge /experience on the entities types involved (example: patient, Clinical finding), their roles and responsibilities and the relationships /associations between different enterprise function and /or entity types. This paper aims at creating a reference data model that will serve as a generic starting point for any new HIS development projects so costs involved in studying and analyzing current state and coming up with gaps analysis and additional requirements can be significantly reduced. The model is Hierarchical in nature that is it is dived into 3 levels of sub models and units so a choice for full or partial implementation can be offered based on the requirements.

CHAPTER 3
System Design

3.1 SYSTEM DESIGN:

3.1.1INTRODUCTION TO UML:

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

1. **Visualizing**

   Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

2. **Specifying**

   Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

3. **Constructing**

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

4. **Documenting**

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

4.2 UML Approach

UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. State chart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

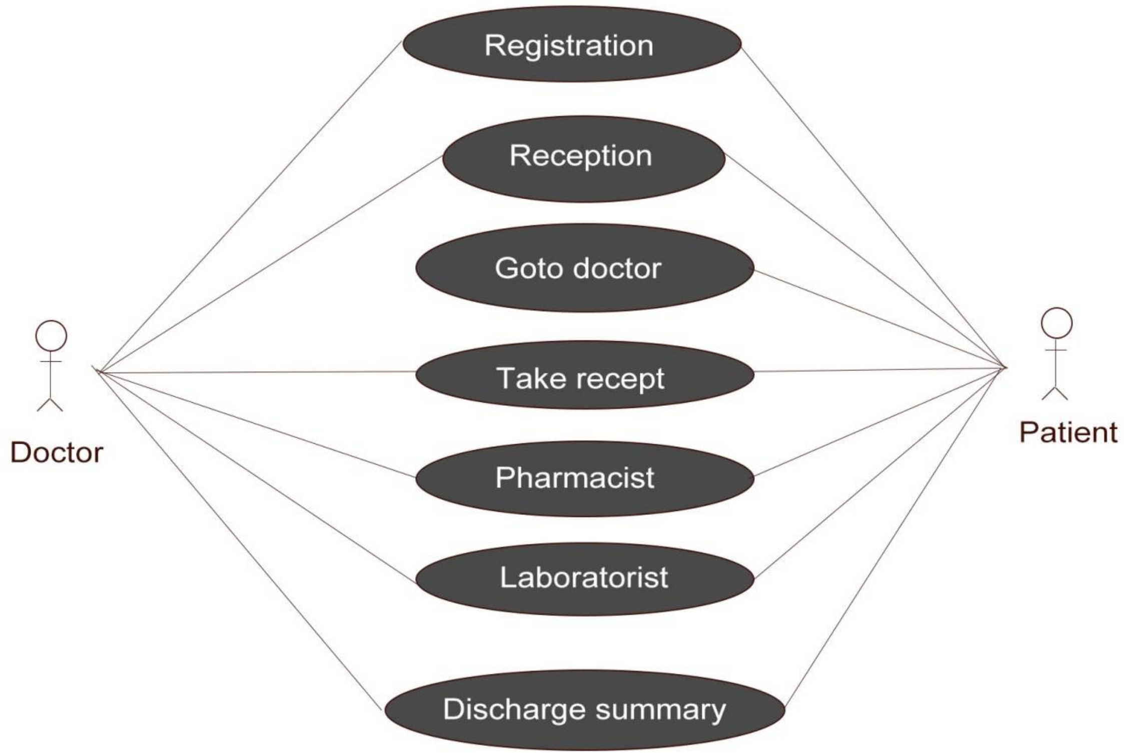
1. USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral 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 modeling languages defined by the unified modeling language (UML) and the systems modeling language (SysML)

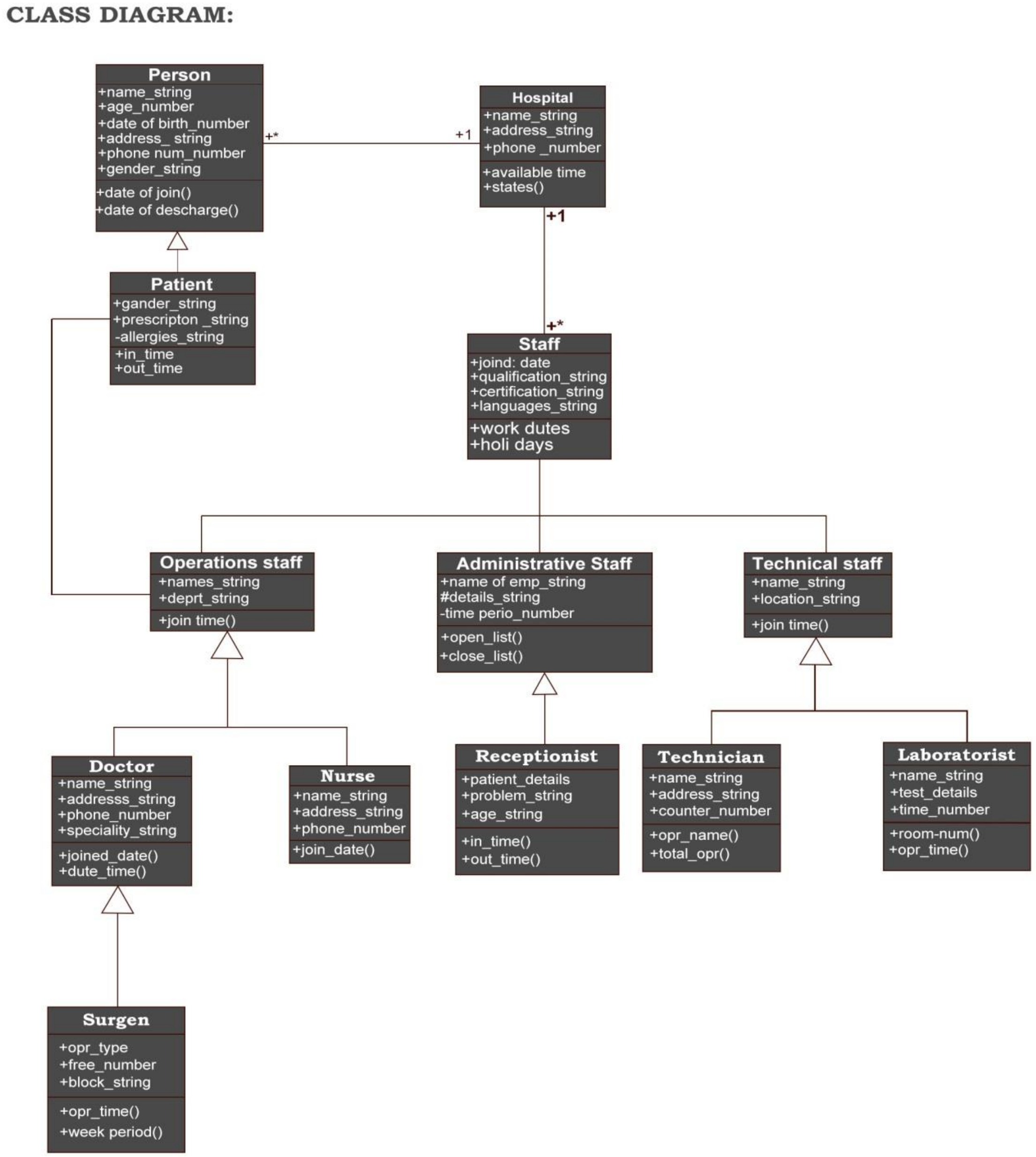
System Design

USE CASE DIAGRAM



2. **CLASS DIAGRAM**:

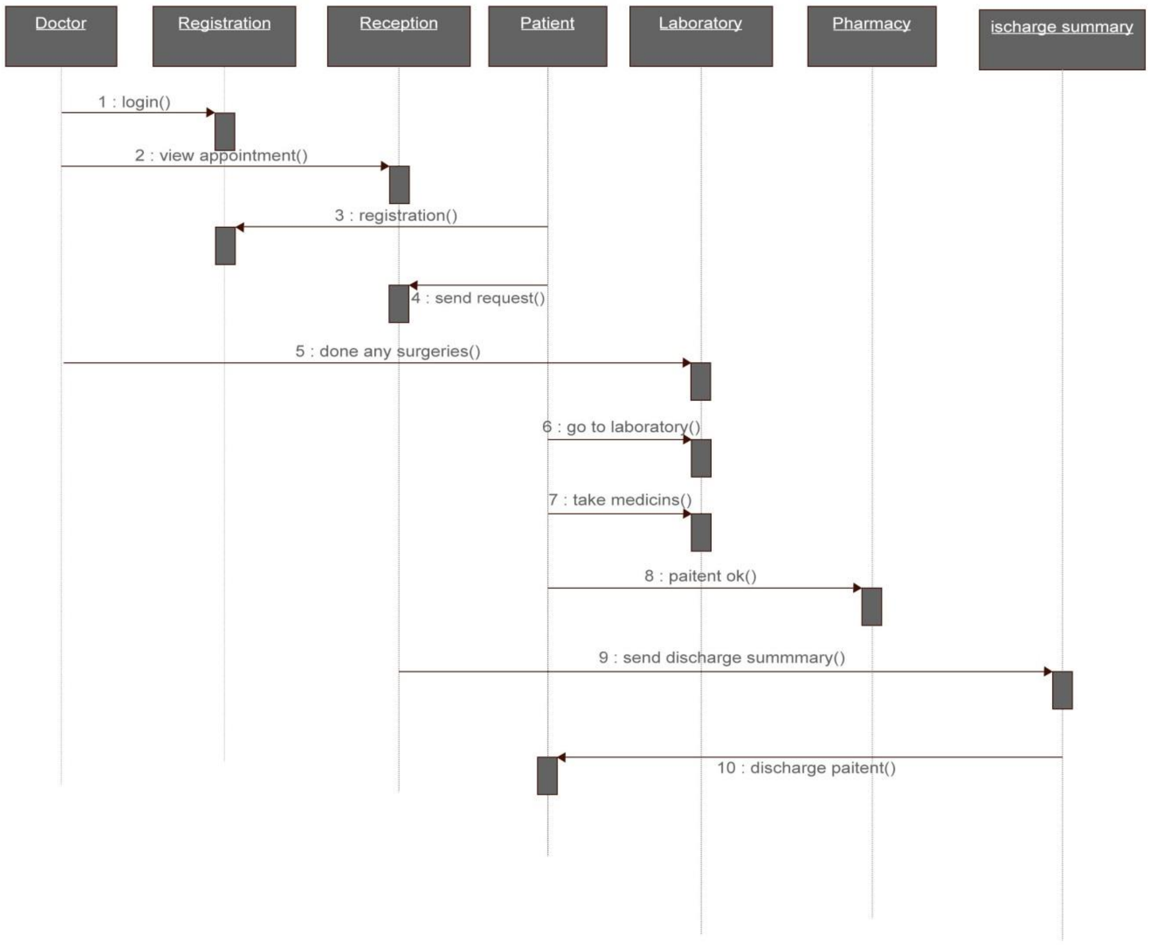
   A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams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



3. 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.

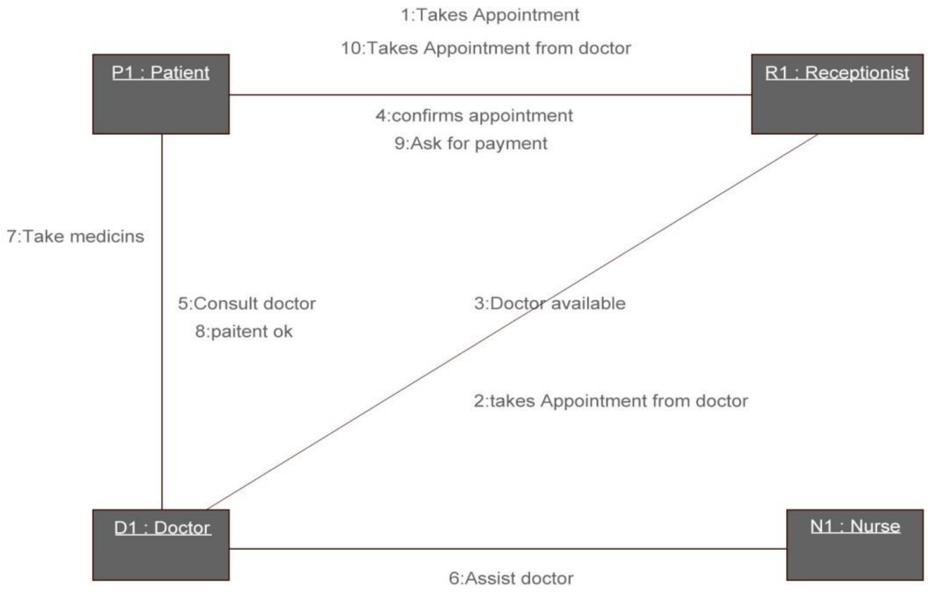
Sequence diagram :



4. COLLABORATION DIAGRAM:

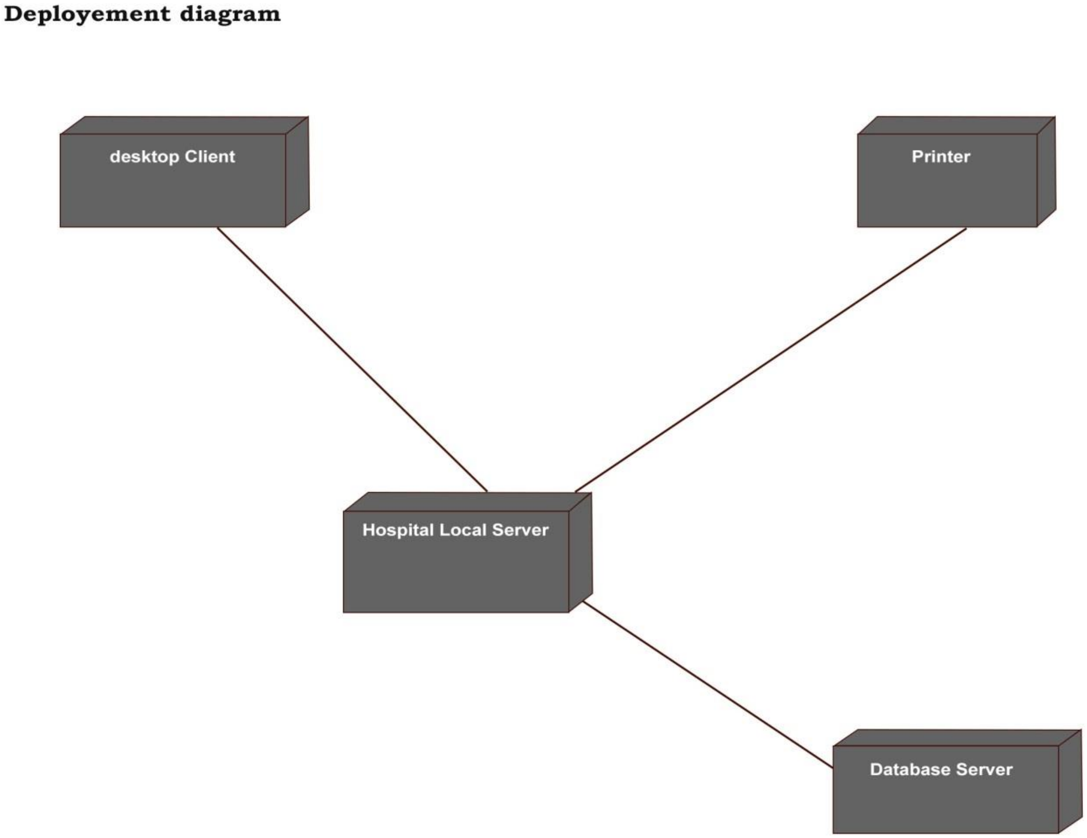
A **Collaboration Diagram** also called a communication diagram or interaction diagram is an illustration of the relationships and interactions among software objects. The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

Collaboration diagram



5. 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.

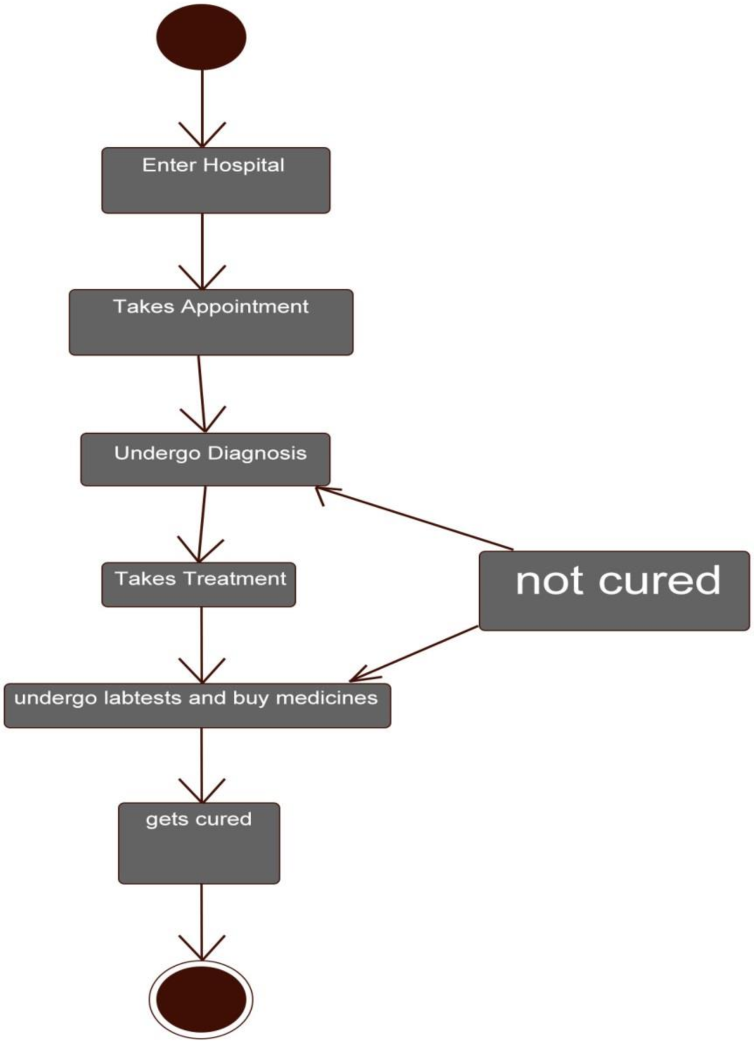


6. STATECHART DIAGRAMS:

The state diagram shows the states of an object and represents activities as arrows connecting the states. The Activity Diagram highlights the activities. Each activity is represented by a rounded rectangle-narrower and more oval-shaped than the state icon. An arrow represents the transition from the one activity to the next. The activity diagram has a starting point represented by filled-in circle, and an end point represented by bull’s eye.

System Design

Statechart diagram



CHAPTER 4
MODULES

MODULES:

The entire project mainly consists of 7 modules, which are

1. Admin module
2. User module (patient)
3. Doctor module
4. Nurse module
5. Pharmacist module
6. Laboratories module
7. Accountant module

1 Admin Module:

- Manage department of hospitals, user, doctor, nurse, pharmacist, laboratories accounts.
- watch appointment of doctors
- watch transaction reports of patient payment
- Bed ,ward, cabin status
- watch blood bank report
- watch medicine status of hospital stock
- watch operation report
- watch birth report
- watch diagnosis report
- watch death report

MODULES

2 User Module (patient):

- View appointment list and status with doctors
- View prescription details
- View medication from doctor
- View doctor list
- View blood bank status
- View operation history
- View admits history. Like bed, ward ICU etc.
- Manage own profile

3 Doctor Modules:

- Manage patient. account opening and updating
- Create, manage appointment with patient
- Create prescription for patient
- Provide medication for patients
- Issue for operation of patients and creates operation report
- Manage own profile

4 Nurse Module:

- Manage patient. account opening and updating
- Allot bed, ward, cabin for patients
- Provide medication according to patient prescription
- Manage blood bank and update status
- Keep record of patient operation, baby born and death of patient
- Manage own profile

MODULES

5 Pharmacist Module:

- Maintain medicine
- Keep records of hospitals stock medicines and status
- Manage medicine categories
- Watch prescription of patient
- Provide medication to prescriptions

6 Laboratories Module:

- Watch prescription list
- Upload diagnostic report
- Preview of report files. like x-ray images, CT scan, MRI reports
- Manage own profile

7 Accountant Module:

- Create invoice for payment
- Order invoice to patient
- Take cash payment
- Watch payment history of patients
- Manage own profile

CHAPTER 5
SYSTEM IMPLEMENTATION

IMPLEMENTATION:

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 it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

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>
```

SAMPLE SOURCE CODE

```
<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   ">
```

SAMPLE SOURCE CODE

```
</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>
```

SAMPLE SOURCE CODE

```
<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>
```

SAMPLE SOURCE CODE

```
</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>
```

SAMPLE SOURCE CODE

```
</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>";
```

SAMPLE SOURCE CODE

```
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">
```

SAMPLE SOURCE CODE

```
<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>
```

SAMPLE SOURCE CODE

```
        <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">
```

SAMPLE SOURCE CODE

```
<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>";
```

SAMPLE SOURCE CODE

```
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">
```

SAMPLE SOURCE CODE

```
 <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>
```

CHAPTER 6
TESTING

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.

TESTING

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.

TESTING

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.

Test Results:

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

Acceptance Testing:

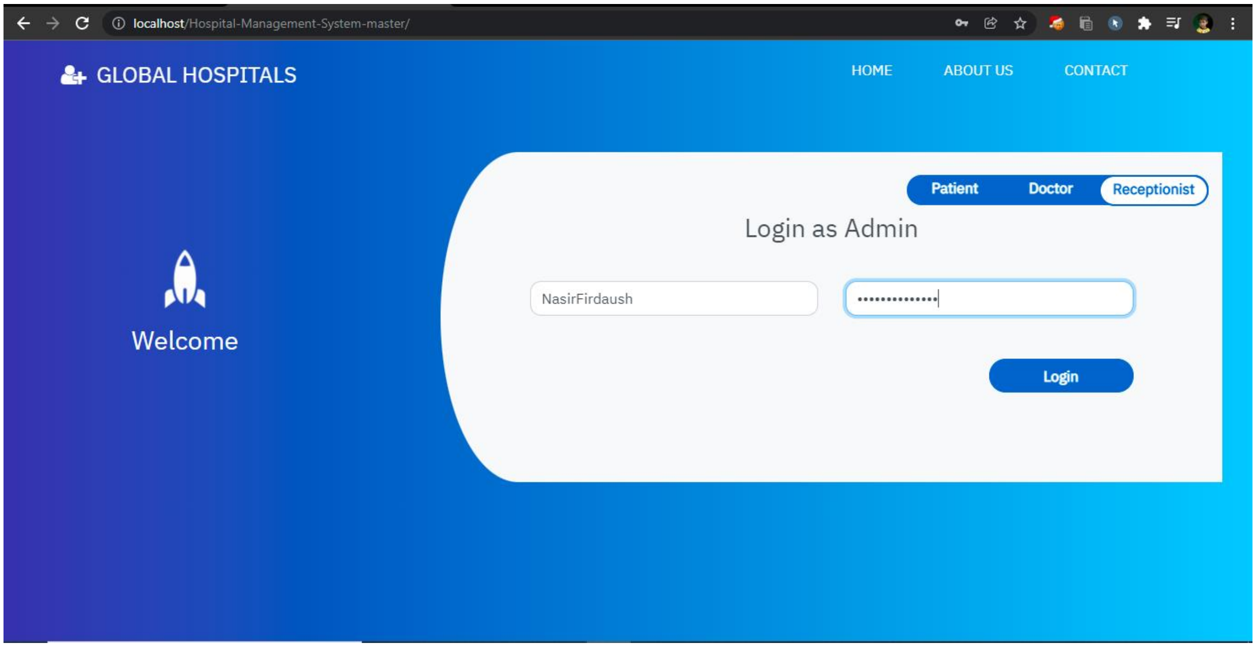
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

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

CHAPTER 7
SAMPLE SCREENSHOTS

ADMIN LOGIN:-



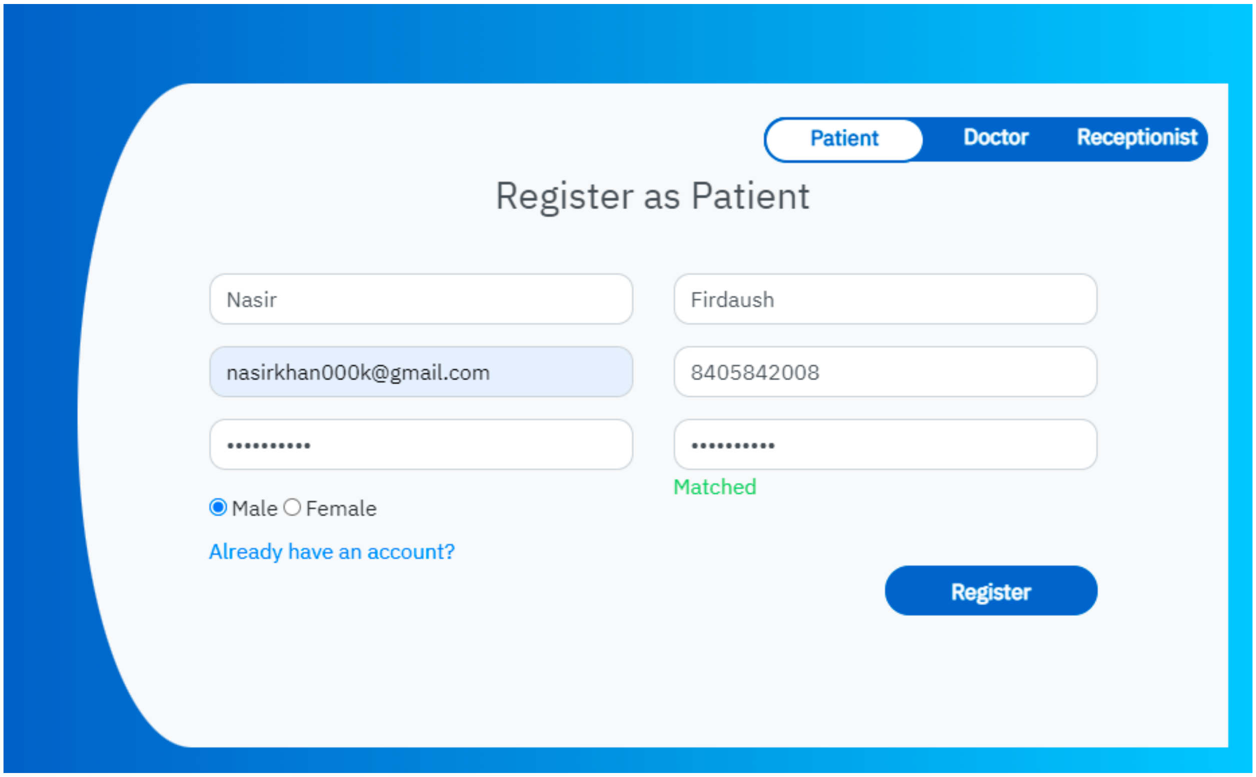
localhost/Hospital-Management-System-master/

Login as Admin

NasirFirdaush

••••••••••••••

PATIENT REGISTRATION:



Register as Patient

Nasir

Firdaush

nasirkhan000k@gmail.com

8405842008

••••••••••

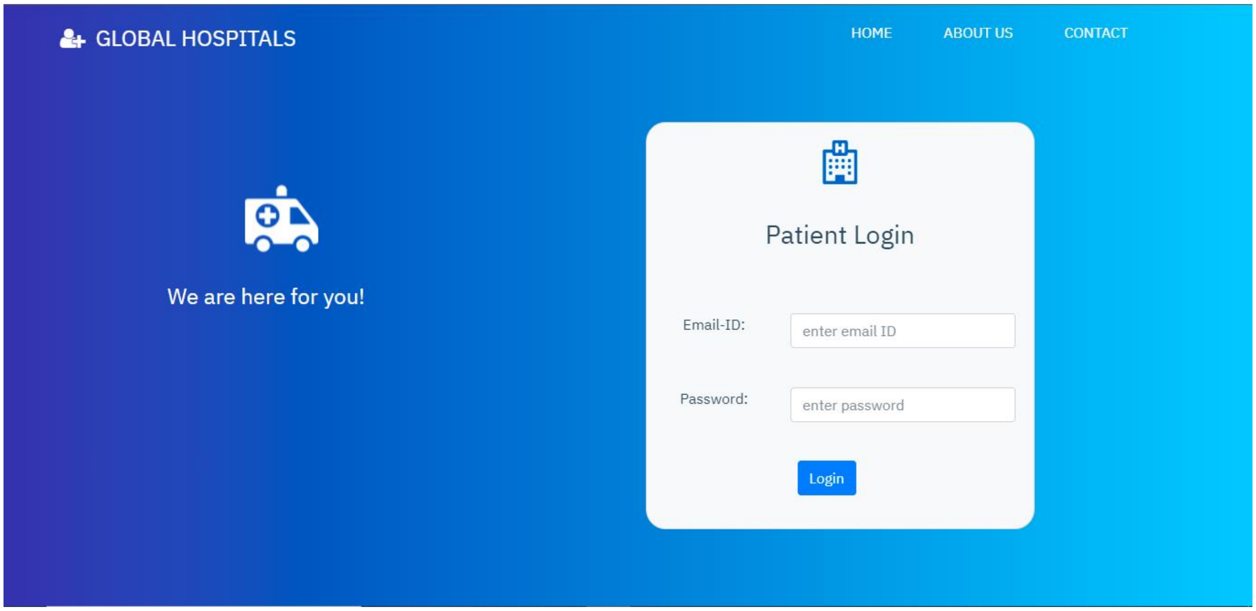
••••••••••

Matched

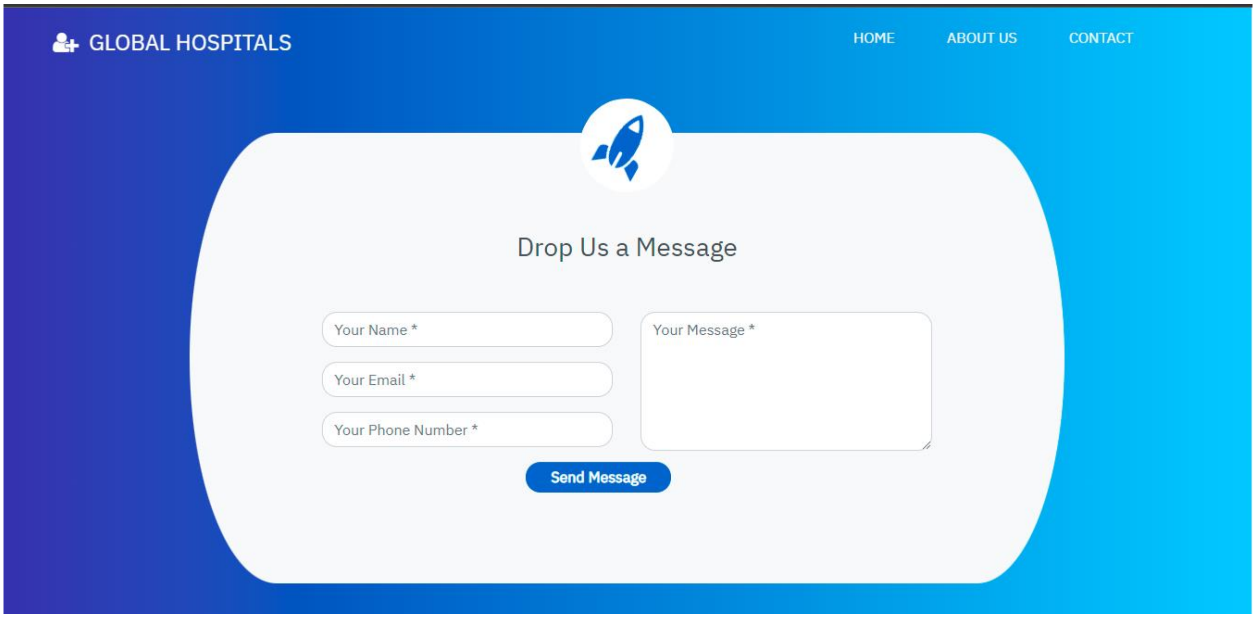
● Male ○ Female

SAMPLE SCREENSHOTS

PATIENT LOGIN:

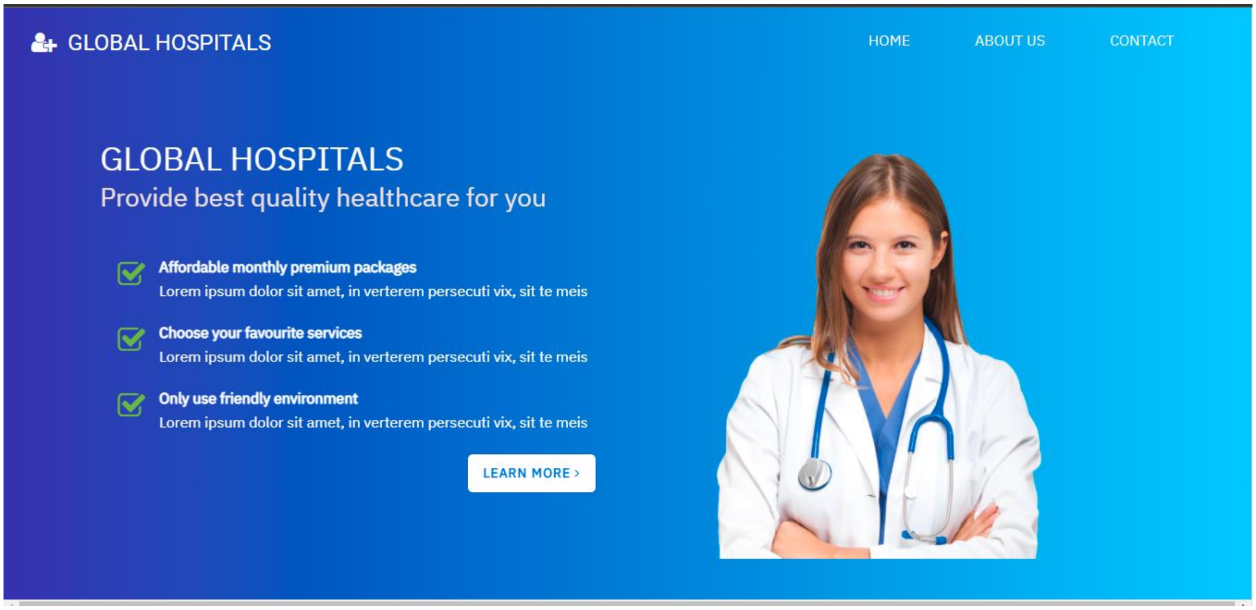


CONTACT:



SAMPLE SCREENSHOTS

ABOUT:

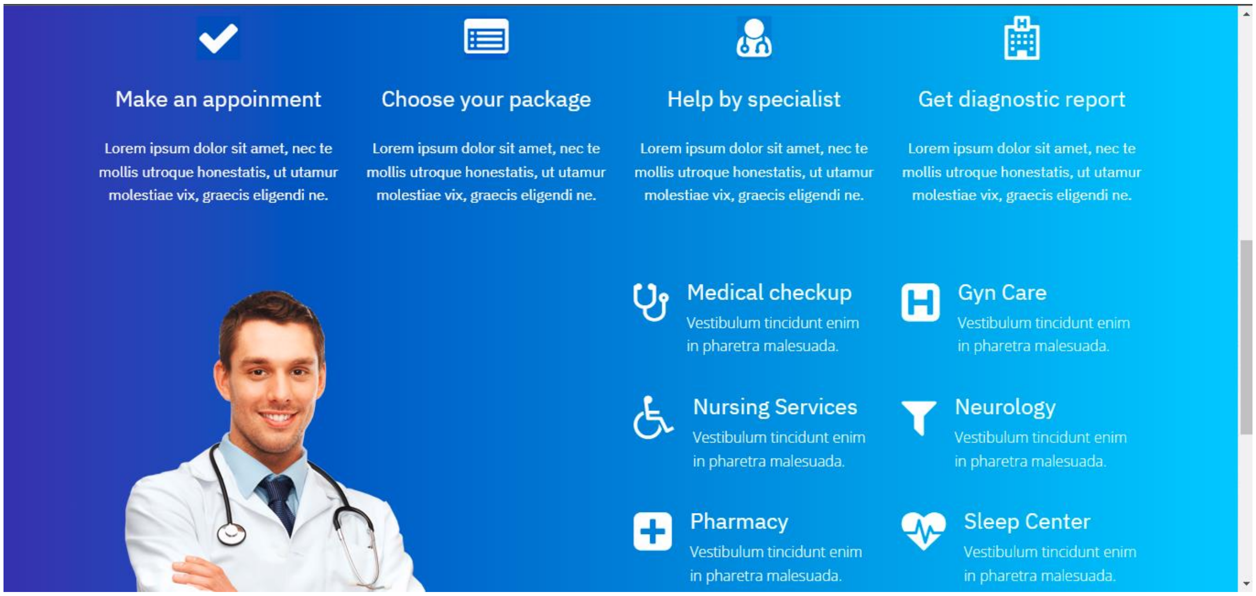


GLOBAL HOSPITALS
Provide best quality healthcare for you

✓ **Affordable monthly premium packages**
Lorem ipsum dolor sit amet, in verterem persecuti vix, sit te meis

✓ **Choose your favourite services**
Lorem ipsum dolor sit amet, in verterem persecuti vix, sit te meis

✓ **Only use friendly environment**
Lorem ipsum dolor sit amet, in verterem persecuti vix, sit te meis



Make an appoinment
Lorem ipsum dolor sit amet, nec te mollis utroque honestatis, ut utamur molestiae vix, graecis eligendi ne.

Choose your package
Lorem ipsum dolor sit amet, nec te mollis utroque honestatis, ut utamur molestiae vix, graecis eligendi ne.

Help by specialist
Lorem ipsum dolor sit amet, nec te mollis utroque honestatis, ut utamur molestiae vix, graecis eligendi ne.

Get diagnostic report
Lorem ipsum dolor sit amet, nec te mollis utroque honestatis, ut utamur molestiae vix, graecis eligendi ne.

Medical checkup
Vestibulum tincidunt enim in pharetra malesuada.

Gyn Care
Vestibulum tincidunt enim in pharetra malesuada.

Nursing Services
Vestibulum tincidunt enim in pharetra malesuada.

Neurology
Vestibulum tincidunt enim in pharetra malesuada.

Pharmacy
Vestibulum tincidunt enim in pharetra malesuada.

Sleep Center
Vestibulum tincidunt enim in pharetra malesuada.

CHAPTER 8
Conclusion

FURTHER SCOPE

1. Free revelation of the desired input.
2. An element of bias might have crept in from the side of the official interviewed. This could also have resulted in some kind of modification of the information divulged.
3. Through an attempt was make to collect information from the best possible source in the company, it was difficult to meet the top officials due to their busy schedules.
4. Most of the analysis and interpretations, made for this report, are based on secondary data obtained. This data could have some inherent mistakes and errors.
5. Finally, although due care has been taken those can be typing and compilation errors in the report itself.

The tasks specified were not well defined because nothing was mentioned regarding validations in the project. Though we gave maximum effort to check the software with different validation tests, a few of them might be present in this version.

- Due to limited time available survey could not be undertaken for intended 20 consumers and thus had to be limited to 10.
- Communication gaps exist between employees and management, as seniors don't share problem with subordinates resulting in violation of psychological contract.
- Poor rewarding system(slow)
- Poor working conditions

CHAPTER 8
Conclusion

The limitations may be many and the magnitude of the influence of these limiting factors may have a bearing on the report, but it in no way alters the ultimate aim of the project and because it's highly USER FRIENDLY, it would be the choice of all kinds of personnel.

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.

References

1- HTML &CSS https://www.w3schools.com/

2- Bootstraphttp://getbootstrap.com/

3- https://stackoverflow.com

4- Abraham Silberschatz, Henry F. Korth and S. Sudarshan “Sixth Edition Database System Conceptsreleased

5- https://books.google.iq/books?id=hxMiygsvOh8C&pg=PA36&lpg=PA36&dq=hms+c%23&source=bl&ots=uh0E6Kov04&sig=ACfU3U23pLOPSMtsRQuQs6ab44a6pqV5Nw&hl=ar&sa=X&ved=2ahUKEwifgb-RlKDqAhUoxKYKHSFsATEQ6AEwCXoECAgQAQ#v=onepage&q=hms%20c%23&f=false

6- https://books.google.iq/books?id=gK3l6LM5DhQC&pg=RA1-PA121&lpg=RA1-PA121&dq=hms+c%23&source=bl&ots=mxo_T96ia1&sig=ACfU3U0vG17DHSzUqTQQKFNrut

7- https://www.youtube.com/watch?v=iqm1Y6fNqbg

8- https://www.edx.org/course/cs50s-introduction-to-computer-science

9- https://www.freecodecamp.org/

10- https://www.codeconquest.com/