স্থায়ী সনদ প্রাপ্ত বিশ্ববিদ্যালয়

City University

...creating a culture of excellence

# CITY UNIVERSITY

## HOSPITAL MANAGEMENT SYSTEM REPORT

**Project Name : HOSPITAL MANAGEMENT SYSTEM**

**Course Title : SYSTEM ANALISIS AND DESIGN**

**Submitted To : Richard Philip**

**Senior Lecturer, Department of CSE**

**City University, Bangladesh**

**Submitted By:**


| | |
|---|---|
| **Name** | **: Sagor Roy Chowdhury** |
| **ID** | **: 171442606** |
| **Program** | **: CSE(Eve)** |
| **Batch** | **: 44th** |

# ABSTRACT

The purpose of the project entitled as “HOSPITAL MANAGEMENTSYSTEM” is to computerize the Front Office Management of Hospital to develop software whichis user friendlysimple, 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.

# INDEX

# 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

## 1.2 Problem Introduction:

**Lack of immediate retrievals: -**

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

**Lack of immediate information storage: -**

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

**Lack of prompt updating: -**

Various changes to information like patientdetails or immunization details of child are difficult to make as paper work isinvolved.

**Error prone manual calculation: -**

Manual calculations are error prone andtake a lot of time this may result in incorrect information. For examplecalculation 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.

**Scope of the Project:-**

1) Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.
2) Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up.
3) Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
4) Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
5) Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can'tremember them at that time.

## 1.3 MODULES:

The entire project mainly consists of 7 modules, which are

- Admin module
- User module (patient)
- Doctor module
- Nurse module
- Pharmacist module
- Laboratories module
- Accountant module

### 1.3.1 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

### 1.3.2 Usermodule (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 admit history. like bed, ward icu etc
- Manage own profile

### 1.3.3 Doctor module:

- 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

### 1.3.4 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

### 1.3.5 Pharmacist module:

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

### 1.3.6 Laboratories module:

- Watch prescription list
- Upload diagnostic report
- Preview of report files. like xray images, ct scan, mri reports
- Manage own profile

### 1.3.7 Accountant module:

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

# CHAPTER 2

# REQUIREMENTSPECIFICATION

## 2.1 INTRODUCTION:

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as(computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

## 2.2 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

RAM : 1 GB

HARD DISK : 80 GB

### 2.3 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 3

# ANALYSIS

### 3.1EXISTING 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 throughout 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.

## 3.2 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.

## 3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

### 3.3.1 Economic Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

### 3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

### 3.3.3 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the

user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### 3.4 SOFTWARE SPECIFICATION

**HTML:**

**HTML** or **Hypertext Markup Language** is the standard markup language used to create web pages.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

**CASCADING STYLE SHEETS** (**CSS**):

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as thelayout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content .

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

**MySQL:**

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web it runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use.It supports standard SQL. MySQL can be compiled on a number of platforms.

The data in MySQL is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically.

**FEATURES OF MySQL:**

Internals and portability:

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms.
- Tested with Purify (a commercial memory leakage detector) as well as with Val grind, a GPL tool.
- Uses multi-layered server design with independent modules.

Security:

- A privilege and password system that is very flexible and secure, and that enables host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

Scalability and Limits:

- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- Support for up to 64 indexes per table (32 before MySQL 4.1.2). Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 767 bytes for **InnoDB** tables, or 1000 for **MyISAM**; before MySQL 4.1.2, the limit is 500 bytes. An index may use a prefix of a column for **CHAR**, **VARCHAR**, **BLOB**, or **TEXT** column types.

**CONNECTIVITY:**

Clients can connect to MySQL Server using several protocols:

- Clients can connect using TCP/IP sockets on any platform.
- On Windows systems in the NT family (NT, 2000, XP, 2003, or Vista), clients can connect using named pipes if the server is started with the --enable-named-pipe option. In MySQL 4.1 and higher, Windows servers also support shared-memory connections if started with the --shared-memory option. Clients can connect through shared memory by using the --protocol=memory option.
- On UNIX systems, clients can connect using Unix domain socket files.

**LOCALIZATION**:

- The server can provide error messages to clients in many languages.
- All data is saved in the chosen character set.

**CLIENTS AND TOOLS**:

- MySQL includes several client and utility programs. These include both command-line programs such as **MySQL dump** and **mysqladmin**, and graphical programs such as MySQL Workbench.

- MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the **MySQL check** client. MySQL also includes **myisamchk**, a very fast command-line utility for performing these operations on **MyISAM** tables.
- MySQL programs can be invoked with the --help or -? Option to obtain online assistance.

**WHY TO USE MySQL:**

- Leading open source RDBMS
- Ease of use – No frills
- Fast
- Robust
- Security
- Multiple OS support
- Free
- Technical support
- Support large database– up to 50 million rows, file size limit up to 8 Million TB

**JAVASCRIPT:**JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

**WHY TO USE JAVASCRIPT**:

JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to specify the behavior of web pages

Example

x = document. get Element ById("demo"); //Find the HTML element with id="demo" x.innerHTML = "Hello JavaScript"; //Change the content of the HTML element

**document.getElementById()** is one of the most commonly used HTML DOM methods.

**OTHER USES OF JAVASCRIPT**:

- Delete HTML elements
- Create new HTML elements
- Copy HTML elements

- In HTML, JavaScript is a sequence of statements that can be executed by the web browser.

**JAVASCRIPT STATEMENTS**:

- JavaScript statements are "commands" to the browser.
- The purpose of the statements is to tell the browser what to do.
- This JavaScript statement tells the browser to write "Hello Dolly" inside an HTML element with id="demo":

  Semicolon;
- Semicolon separates JavaScript statements.
- Normally you add a semicolon at the end of each executable statement.
- Using semicolons also makes it possible to write many statements on one line.

**JAVASCRIPT CODE:**

- JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written.
- This example will manipulate two HTML elements:
- Example
- document.getElementById("demo").innerHTML="Hello Dolly";
  document.getElementById("myDIV").innerHTML="How are you?";

**JAVASCRIPT PROPERTIES:**

- Properties are the values associated with a JavaScript object.
- A JavaScript object is a collection of unordered properties.
- Properties can usually be changed, added, and deleted, but some are read only.

**PHP:**

**WHAT IS PHP?**

- PHP is an acronym for "PHP Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP costs nothing, it is free to download and use

**WHAT IS PHP FILE?**

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

**WHAT CAN PHP DO?**

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

**WHY PHP?**

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net

# CHAPTER 4

# DESIGN

## 4.1 SYSTEM DESIGN:

### 4.1.1INTRODUCTION 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:

- Visualizing
- Specifying
- Constructing
- Documenting

**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.

**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.

**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.

**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:

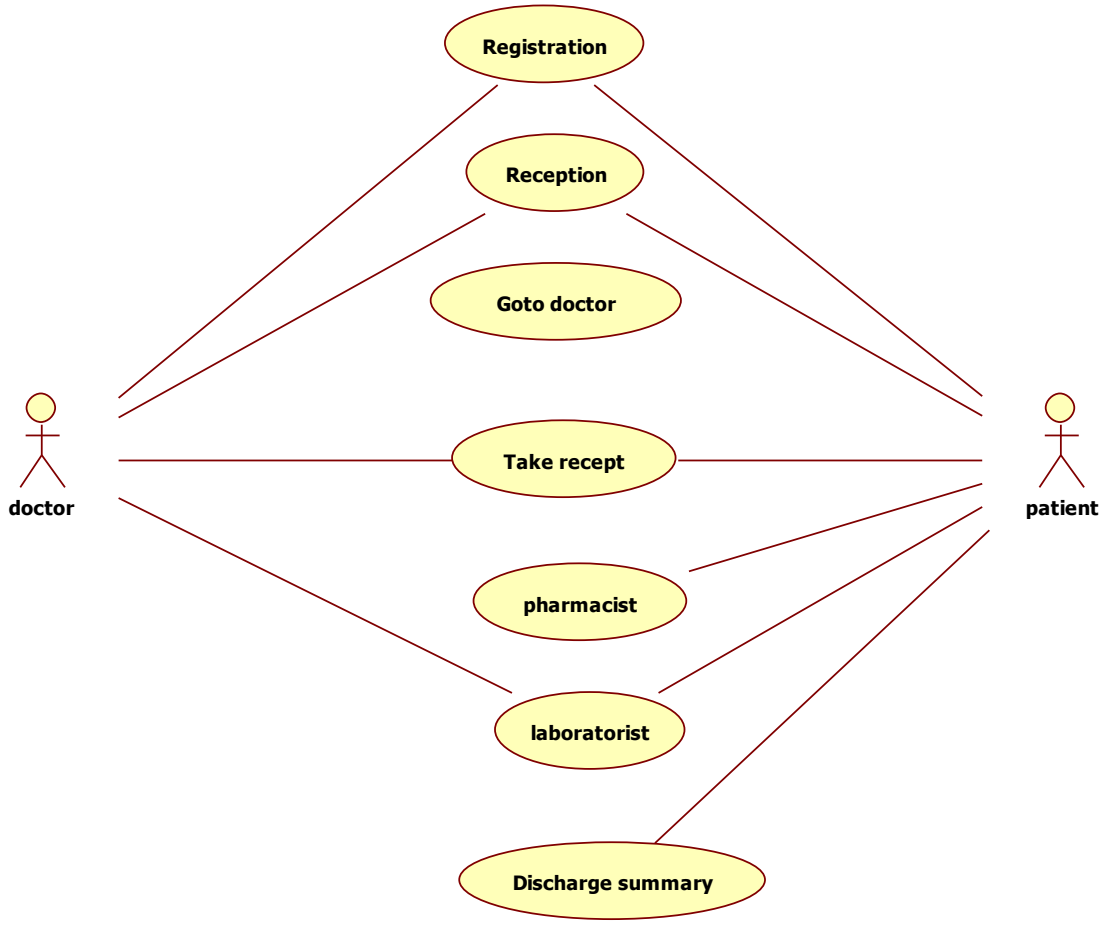
1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. State chart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

## USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language(UML) is atype 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, theirgoals (represented as use cases),and any dependencies between those use cases.

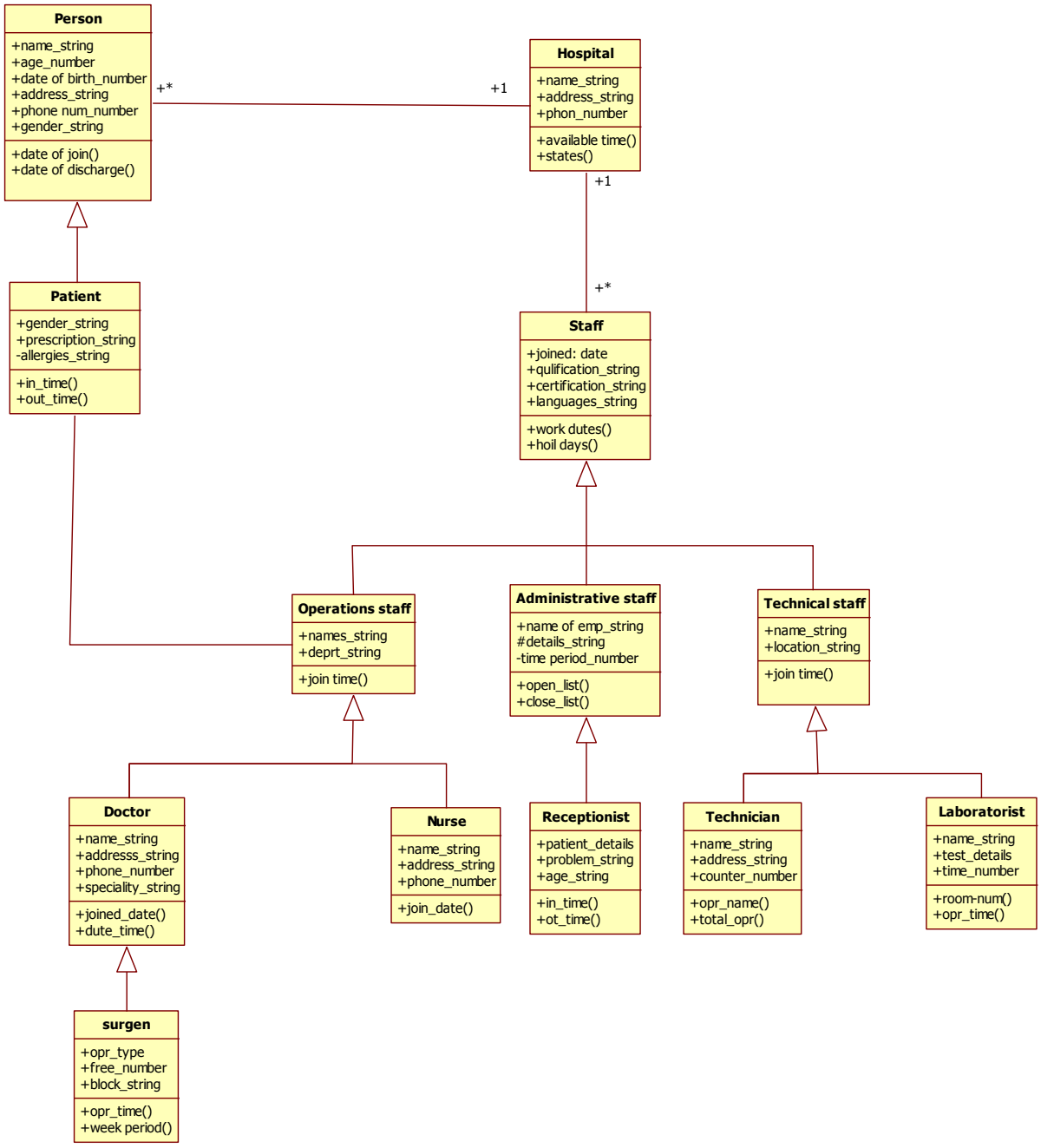
Use case diagrams are formally included in two modeling languages defined by the OMG:theunfied modeling language(UML) and the systems modeling language(sysML)

## Use case diagram of our project:




## Class Diagram:

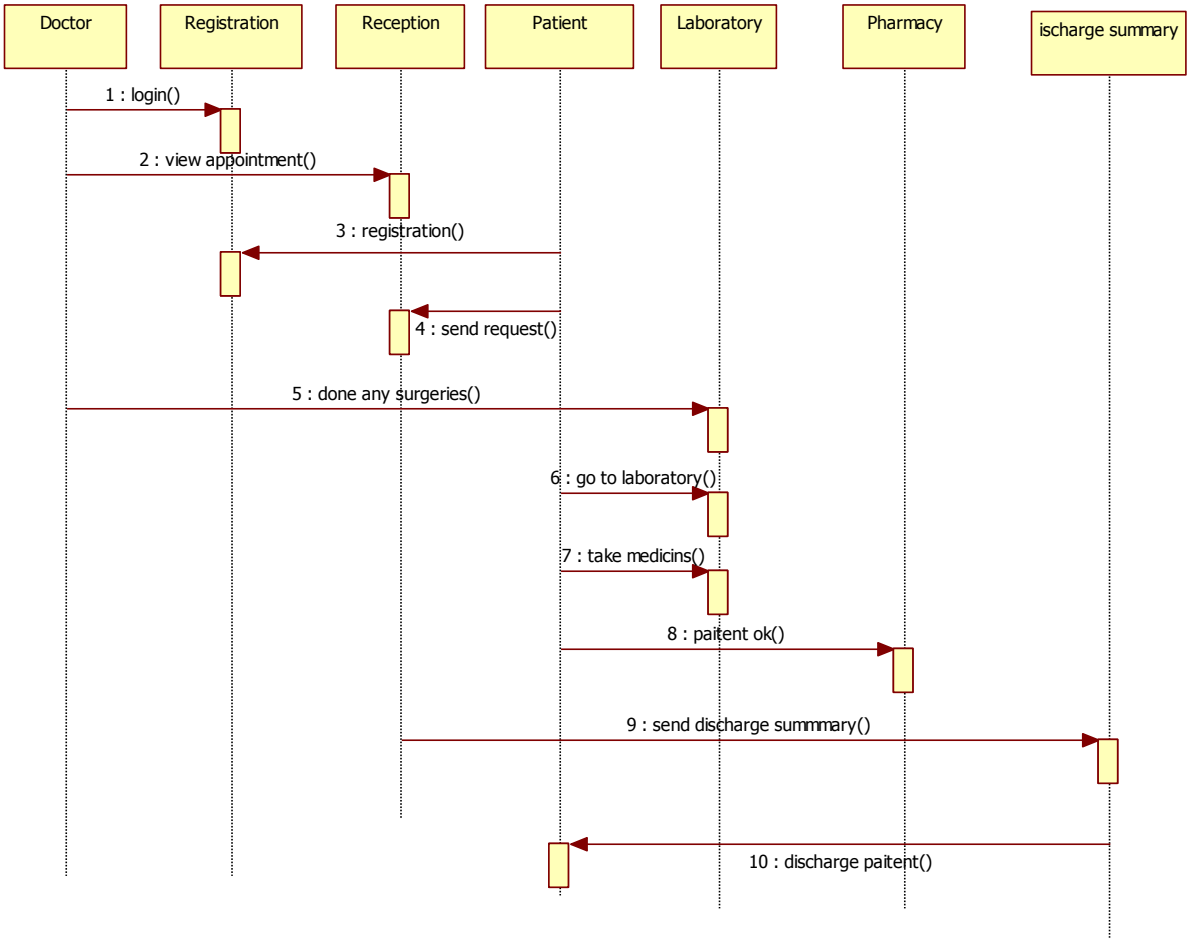
A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams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


Person
+name_string
+age_number
+date of birth_number
+address_string
+phone num_number
+gender_string
+date of join()
+date of discharge()
+*
+1
Hospital
+name_string
+address_string
+phon_number
+available time()
+states()
+1
+*
Patient
+gender_string
+prescription_string
-allergies_string
+in_time()
+out_time()
Staff
+joined: date
+qulification_string
+certification_string
+languages_string
+work dutes()
+hoil days()
Operations staff
+names_string
+deprt_string
+join time()
Administrative staff
+name of emp_string
#details_string
-time period_number
+open_list()
+close_list()
Technical staff
+name_string
+location_string
+join time()
Doctor
+name_string
+addresss_string
+phone_number
+speciality_string
+joined_date()
+dute_time()
Nurse
+name_string
+address_string
+phone_number
+join_date()
Receptionist
+patient_details
+problem_string
+age_string
+in_time()
+ot_time()
Technician
+name_string
+address_string
+counter_number
+opr_name()
+total_opr()
Laboratorist
+name_string
+test_details
+time_number
+room-num()
+opr_time()
surgen
+opr_type
+free_number
+block_string
+opr_time()
+week period()

## Sequence diagram:

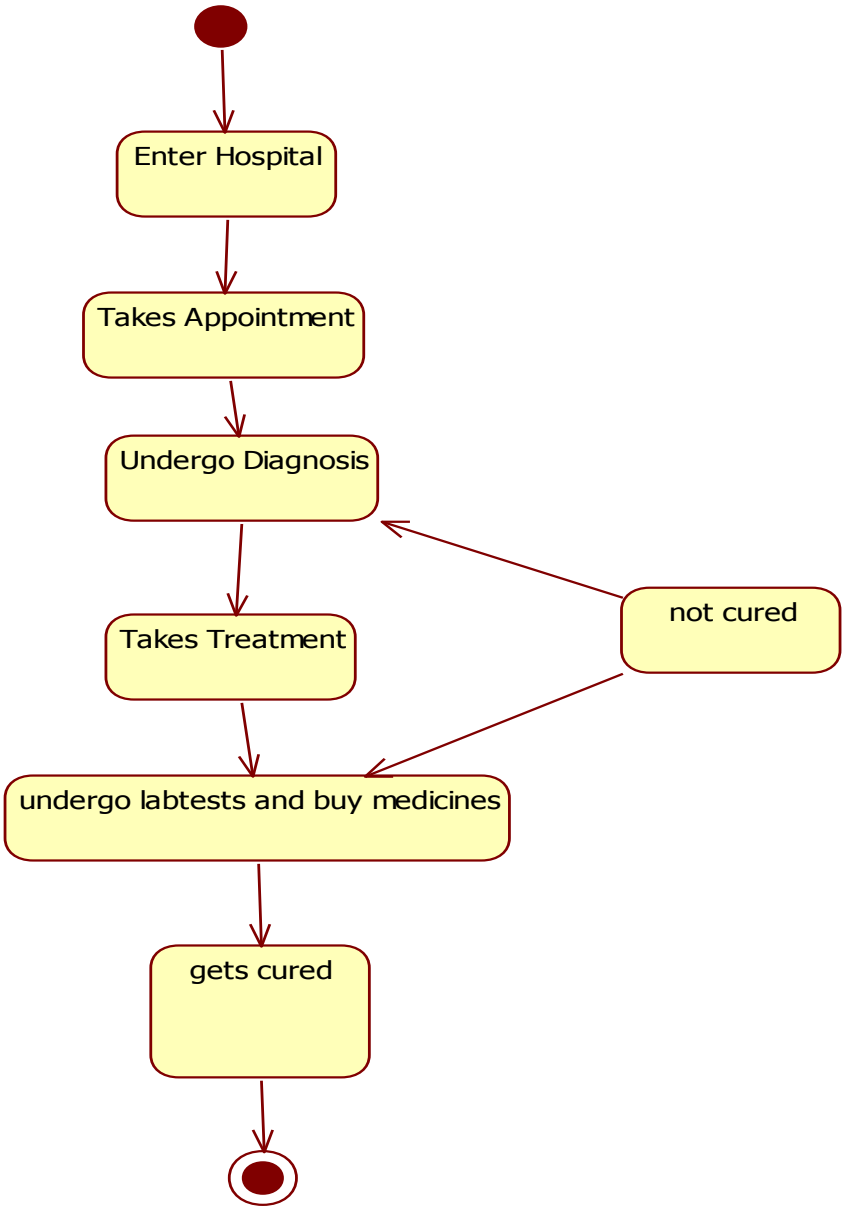
A **Sequence Diagram** is an interaction diagram that emphasis the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning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

than a decade old although




it has been refined as modeling paradigms have evolved.

ACTIVITY DIAGRAM:



# CHAPTER 5

# SYSTEM IMPLEMENTATION

## 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 it's 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>specia
lity</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>
```

# CHAPTER 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.

## 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.