

# **CHAPTER 1**

---

## **INTRODUCTION**

## **1.1 Introduction**

This manuscript is proposing about all the features and dealings to develop the system. Especially it is containing details about objectives, possibility, plan replica, primary and functional requirements, database model and as a final point coverage and analyzing the mechanisms. Once analyzing the mechanisms of the task that would be performed, the following point is to consider the problem and understand its framework. Hospital management system is a project which aims in developing a computerized system to maintain all the daily work of hospital. This project has many features which are generally available in normal hospital management systems like facility of user and a facility of patients. Admin is the most powerful user of the system [1].

## **1.2 What is Hospital Management System?**

Hospital are the essential part of our lives, providing the best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma streets etc. It is necessary for the hospital to keep track of its day-to-day activities and records of its patients, doctors, nurses, pharmacists, accountant, ward boy and other staff personal that keep the hospital running smoothly and successfully. But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and time-consuming process observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone .It is also not economically and technically feasible to maintain these records on paper [3].

## **1.3 Background of the project**

A hospital is a place where patient come up for general diseases. Our project ‘Hospital Management System’ includes registration of patients, storing their details into the system and also computerized billing in the pharmacy and lab. 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 the patient using the id. The advanced hospital management system can be entered using a username and password .It is accessible either by and 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 the personal use and makes the data processing very fast. Hospital Management System is designed for hospital, 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[2].

## 1.4 Objective of the project

The project “Hospital Management System” is aim to develop to maintain the day-to-day state of admission/discharge of patient, List of doctors (consultants), list of medicines, bills etc.

The following main objective of the Hospital management system:

- To computerize all regarding patient details and hospital details.
- Keeping patient-care as utmost priority.
- Scheduling the appointment of patient with doctor (consultant) to make it convenient for both.
- Scheduling the services of specialized doctor (consultant) and emergency properly so that facilities provided by Hospital are fully utilized in effective and efficient manner.
- If the medical store issues medicines to patients, it should reduce the stock status of the medical store and vice -versa.
- It should be able to handle the test reports of patients conducted in the pathology lab in the hospital.
- The inventory should be updated automatically whenever a transaction is made.
- The information of the patients should be kept up to date and their record should be kept in the system for hospital purposes.

## 1.5 Scope of the Project

Hospitals currently use a manual system for the management and maintenance of critical information. Few of them are:

- This system design to facilitate administrator, doctor, patient, nurse and other staff.
- All of the information must be managed in an efficient and cost wise.
- Information typically involves patient and medical history, staff information, room and ward scheduling, operation theater scheduling, bill pay and various facilities.
- Doctor will add patient, proper prescription, provide good treatment etc.
- To be noticed on the scope of the system that this system is a Hospital Management System[1].

## 1.6 Overview of Document

In Chapter 1 I talk about the basic things of my project report, what is hospital management system, scope of this project, objective of this document etc. I also give an overall preview of this document in this chapter.

In Chapter 2 I describe how we analysis the system model, which model we follow to do this project. According to the software development life cycle we follow the waterfall model for complete our project. For Chapter 3 this document describes the requirements gathering procedure and its constraints. Methodological explanation of the project battered near peripheral

viewers. This section includes information such as data requirements, functional requirements, and an over-all narrative of the system and its interaction with users from the perspective of the client.

The system design parts designed by unified model language, we fully document this part to understand easily for the users; and it's describe in Chapter 4. In Chapter 5 we discuss and give some table structures and screenshots of our system for the form design that helps client or user to use this system easily.

Maintenance procedure of this project has been described in Chapter 6. We discuss about the activities and problem phases of maintenance. Finally at the end in Chapter 7 we give the conclusion of this document and talk about limitations and future plans what we will implement in future we hope so.

## **CHAPTER 2**

---

### **SOFTWARE DEVELOPMENT LIFE CYCLE**

#### **2.1Software Development Life Cycle**

SDLC abbreviate as Software Development Life Cycle. The Software Development Life Cycle is a procedure that guarantees good software is constructed. Every segment in the life cycle has its own method and deliverables that feed into the following stage. There are basically five phases initiate with the analysis and requirements gathering and ending with the implementation [4].

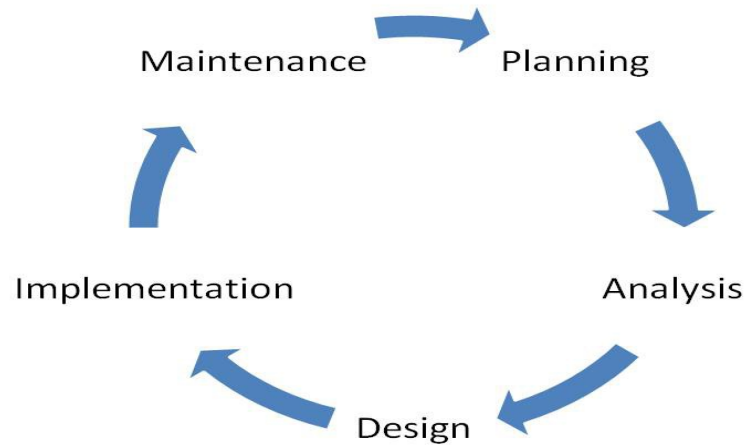


Figure2.1.SDLC Phases

## 2.2SDLC Models

The progress replicas are the several procedures or rules that are being selected for the development of the project contingent on the project's intentions and goals. There are many development life cycle replicas that have been developed in order to achieve different obligatory objectives. The representations specify the various stages of the procedure and the order in which they are carried out. There are various Software progress models or policies. They are as follows [4]:

1. Waterfall model
2. V model
3. Incremental model
4. RAD model
5. Agile model
6. Iterative model
7. Spiral model

## 2.3 Scrutinizing Our Project Model

Selecting accurate model for developing of the software invention or request is very significant. Founded on the model the expansion and testing processes are accepted out. As Waterfall Model is more traditional and easy to gather requirements and analyzing system, so we choose this

model according to complete this project. This technique works well for big projects that may take numerous months to progress.

### 2.3.1 Waterfall Model

The waterfall model is a successive design procedure, used in software advance processes, inwhich development is seen as graceful steadily upwards through the stages of beginning, opening, analysis, design, building, testing, implementation and maintenance [5]. It is also referred to as a linear-successive life cycle model. It is very humble to recognize and custom. In a waterfall model, each stage must be completed fully before the next stage can begin. This type of model is essentially charity for the plan which is small and there are no inexact necessities. At the end of each stage, a review takes place to control if the plan is on the right path and whether or not to endure or abandon the project. In this model the testing starts only after the development is comprehensive. In waterfall model stages do not edge [6].

### 2.3.2 Waterfall Model Design

Waterfall method was major SDLC Model to be used usually in Software Engineering to confirm success of the project. In "The Waterfall" methodology, the entire route of software development is alienated into discrete segments. In Waterfall model, typically, the consequence of one phase acts as the input for the next segment successively [7]. Ensuing is an illustrative representation of dissimilar periods of waterfall model:

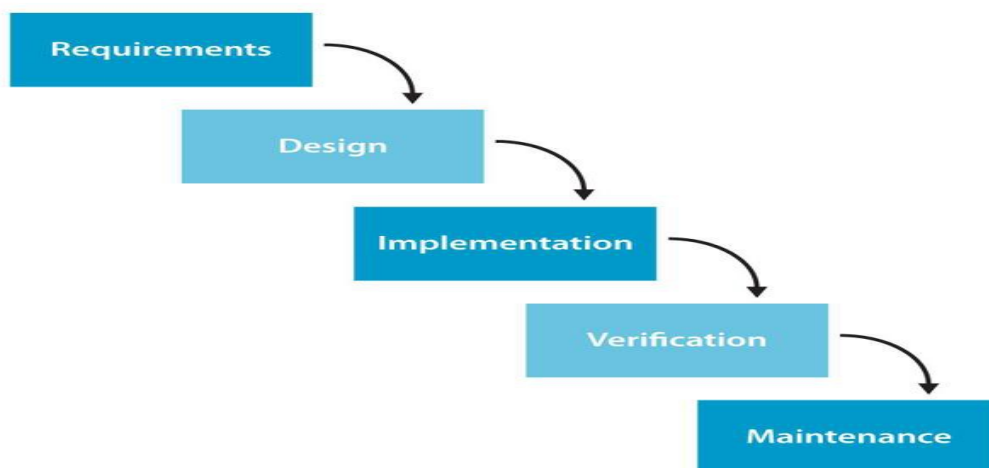


Figure2.2. Waterfall Model

### 2.3.3 Waterfall Model Phases

This segment is acute to the successfully complete the project. Hopes need to be fleshed out in inordinate part and familiar. This is an iterative procedure with much statement taking place

between investors, end customers and the project side. The following techniques can be used to gather supplies [8]:

1. **Requirement gathering and analysis:** Business supplies are assembled in this stage. This segment is the main attention of the project managers and stake holders. Meetings with managers, stake holders and customers are held in order to control the requirements like; that is going to habit the scheme? How will they use the scheme? What statistics should be input into the scheme? What facts should be output by the organization? These general questions can get answer through the requirement gathering time.
2. **Design:** In this segment the scheme and software plan is prepared from the obligation stipulations which were deliberate in the major phase. System Design helps in stipulating hardware and system requirements and also helps in crucial overall system architecture. The system design conditions serve as input for the following stage of the perfect. In this phase the testers comes up with the [test strategy](#), where they reference what to trial, how to trial.
3. **Implementation:** On receiving system design documents, the work is alienated in units and actual coding is started. Since, in this stage the code is formed so it is the main attention for the developer. Coding is the longest period of the software development life cycle.
4. **Testing:** After implementation developed it is tested through the necessities to make sure that the product is truly answering the requests addressed and gathered during the necessities phase. Through this period all sorts of [functional testing](#) like [unit testing](#), [integration testing](#), [system testing](#), [acceptance testing](#) are done as well as [non-functional testing](#) are also complete.
5. **Deployment:** After effective testing the product is organized to the customer for their use. As soon as the product is assumed to the patrons they will first do the [beta testing](#). If any changes are required or if any bugs are caught, then they will crash it to the engineering team. Once those variations are made or the [bugs](#) are fixed then the final placement will happen.
6. **Maintenance:** Once when the clients starts using the advanced system then the actual glitches comes up and needs to be answered from time to time. This procedure where the care is taken for the advanced artifact is known as maintenance

## 2.4 Advantages and Disadvantages of Waterfall Model

### 2.4.1 Advantages of Waterfall Model:

- This model is simple and easy to understand and use.
- It is easy to achieve owing to the inflexibility of the model – each phase has specific deliverables and an assessment procedure.
- In this model periods are treated and finished one at a time. The stages of this model never intersection.



- Waterfall model works well for minor projects where necessities are very well understood [5].

### **2.4.2 Disadvantages of Waterfall Model:**

- Once a request is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the idea stage.
- Working software cannot be produced pending late through the life cycle.
- High quantities of risk and doubt.
- Poor model for long and ongoing projects.
- Not appropriate for the projects where necessities are at a reasonable to high risk altering [5].

## **CHAPTER 3**

---

### **REQUIREMENT GATHERING/ANALYSIS**

#### **3.1 Requirement Analysis**

Requirements analysis in systems engineering and software engineering, includes those tasks that go into decide the needs or circumstances to meet for a new or changed artifact, taking account of the probably differing supplies of the numerous stakeholders, such as receivers or customers. Requirements analysis is serious to the achievement of a growth venture. Requirements must be

actionable, assessable, testable, connected to recognized business needs or chances, and defined to a near of detail adequate for system design. It can be functional and nonfunctional [9].

## **3.2 System Requirements**

Our system can be used in Windows XP, Windows 7, and Windows 8 with 32 bit operating system and also supported for other platform such as Linux and OS X.

- For Windows XP based computers, a 486 / 66 MHz or higher processor with 8 MB.
- For Windows 7 and Windows 8 based computers, higher processor with 2 GB ram.

## **3.3 Software and Hardware Requirements**

### **3.3.1 Software Requirements:**

- Web server software: Apache Tomcat, XAMPP.
- Server side scripting tool: PHP.
- Database tools: MYSQL DBMS.
- Compatible operating system: Windows
- Front end design tool: Html,Css, Java Script, CodeIgniter framework.

### **3.3.2 Hardware Requirements:**

- Hardware recommended by all the software needed.
- RAM: 256 MB or more
- Hard Drive: 10 GB or more
- Communication hardware to serve client requests.

## **3.4 User Requirements**

To deliver the best service to the users we tried to find out the users necessities which are below:

### **Administrator Aspect:**

- Monitoring the whole system from admin panel.
- Taking back up of the database.
- Creating, deleting and modifying the records.
- Add users for the admin panel.
- Add doctor,patients,nurse and other staff.
- Keeping the patient's record like their details.
- Organizing their member registration system.
- Approve the notice to post.

### **Doctor Aspect:**

- Signing in and signing up to the system.
- Changing their password.
- Resetting forgot password.

#### **Patient Aspect:**

- Signing in and signing up to the system.
- Changing their password.
- Resetting forgot password.

#### **Nurse Aspect:**

- Nurses strive to achieve the best possible quality of life for their patients, regardless of disease or disability.
- Nurses use clinical judgment to protect, promote, and optimize health for families, communities, and populations.
- Nurses can work full- or part-time, and many work on a per diem basis or as traveling nurses.
- Monitoring the transaction system.

## **3.5 Functional Requirements**

The administrator module is to monitoring the whole system. They keep the patient's information. Username is provided when they register. The system must only allow user with valid user name and password to enter. The system performs authorization process which decides what user level can the user must be able to logout after they finished using system.[10].

## **3.6 Nonfunctional Requirements**

In [systems engineering](#) and [requirements engineering](#), a non-functional requirement is a [requirement](#) that stipulates standards that can be used to judge the process of a system, fairly than exact performances. They are compared with [functional requirements](#) that describe exact performance or functions. The strategy for applying functional requirements is detailed in the [system design](#). The idea for applying non-functional requirements is detailed in the [system architecture](#). [11].

## **3.7 Business Requirements**

A business rule is whatever that imprisonments and implements business rules and performs. A rule can apply business policy, make a choice, or conclude new data from existing data. This comprises the rules and rules that the System users should stand by. This contains the price of the project and the discount proposals provided. The users should elude unlawful rules and procedures [9].

### **3.8 Data and Category Requirements**

There are dissimilar classes of users namely patients, doctor, nurse, etc. Depending upon the category of user the access rights are obvious. It means if the user is an administrator then he can be able to adjust the data, delete, add etc. All other users except the Hospital only have the rights to save the info about database. The database stores the details of all the hospitals and doctors along with their specializations and consulting time slots[12]. At once a person get access to this application all his previous and present medical records will be stored to the database. It also provides a 24 hours ambulance information facility. Doctors logging in should be able to update their patients' medical records and access their previous medical records.

# **CHAPTER 4**

---

## **SYSTEM DESIGN (UML)**

### **4.1 Use Case**

In software and systems engineering, a use case is a list of act or event periods, normally defining the relations between a role and a system, to reach a goal [13].



Figure 4.1. Use-Case

### 4.1.1 Elements of Use Case Diagram

#### Actor:

An actor in the [Unified Modeling Language](#) (UML) stipulates a part played by a user or any other scheme that relates with the subject [14].



Figure 4.2. Actor

#### Association:

The association is the relation that is drawn between an actor and a use case. It specifies which actors cooperate with the system to complete various tasks [15].

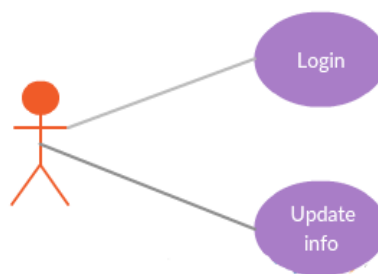


Figure 4.3 Association in Use-Case

#### Includes:

Use contains link to display that one use case includes the task defined by another use case. For example, at the time of login the system verify the username and password; so it has included. Sometimes the word "Uses" is used in its place of "Includes" [15].

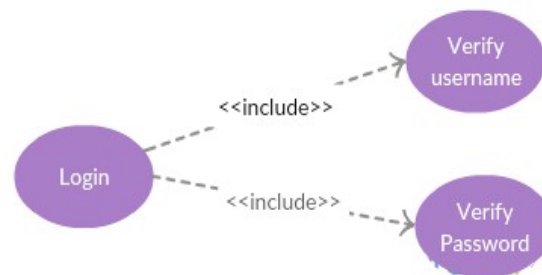


Figure 4.4. Includes Relation in Use-Case

## 4.1.2 Actor Depiction

Actors are exterior entities that cooperate with the structure. Actor pledges system activities for the determination of finishing some task. Actors in this project are as follows:

**Admin:** Monitor the system, add doctors, add patients, add others staff etc.

**Lab Technician:** Saving different lab report information. System can generate lab report for a particular test if relevant information is present in the system.

**Accountant:** Ensure the timely reporting of all monthly financial information.

## 4.1.3 Use-Case Diagram for Hospital Management

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



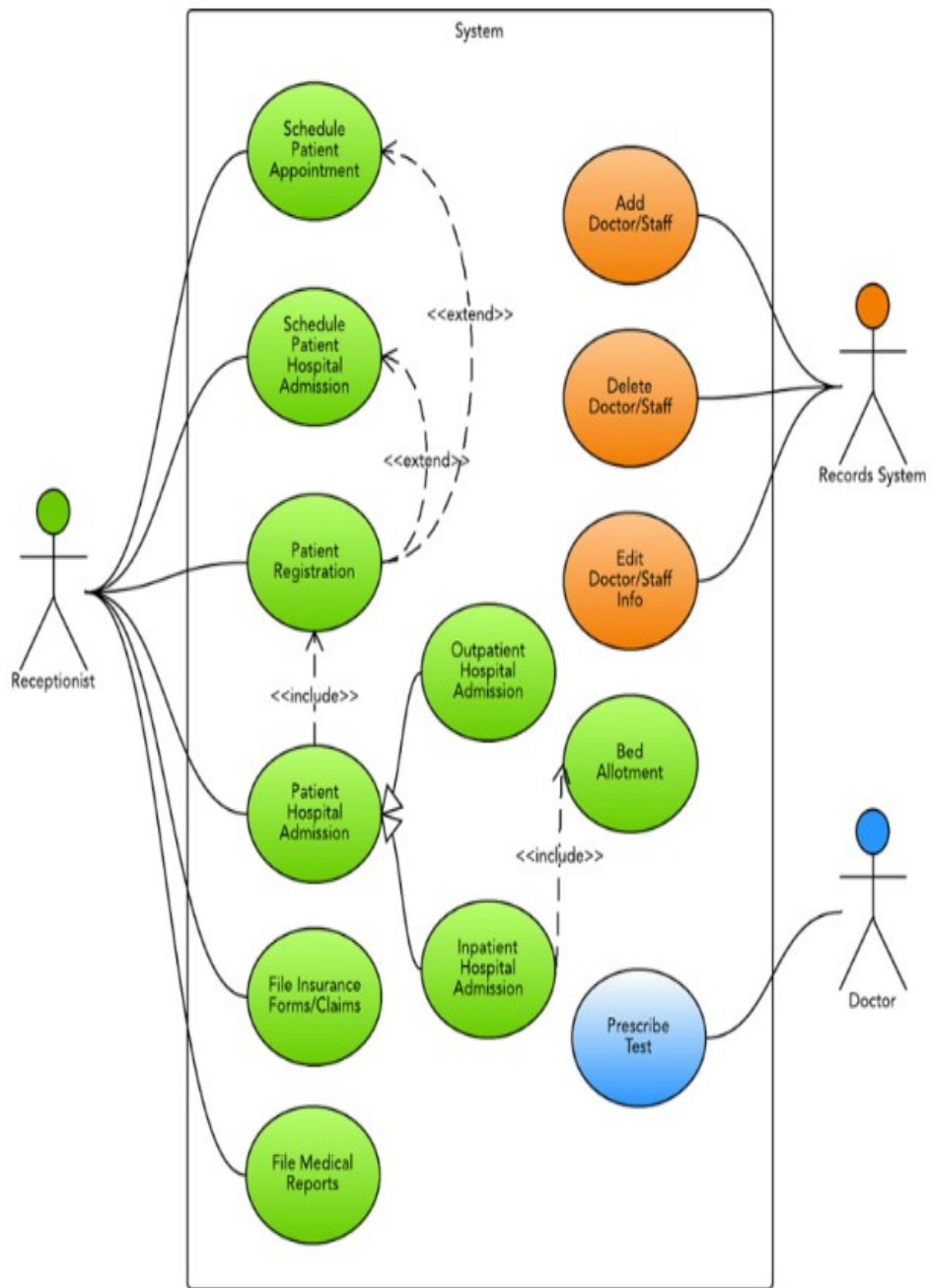


Figure:4.5 Use-Case Diagram for Hospital Management

## 4.2 Class Diagram of Hospital Management System

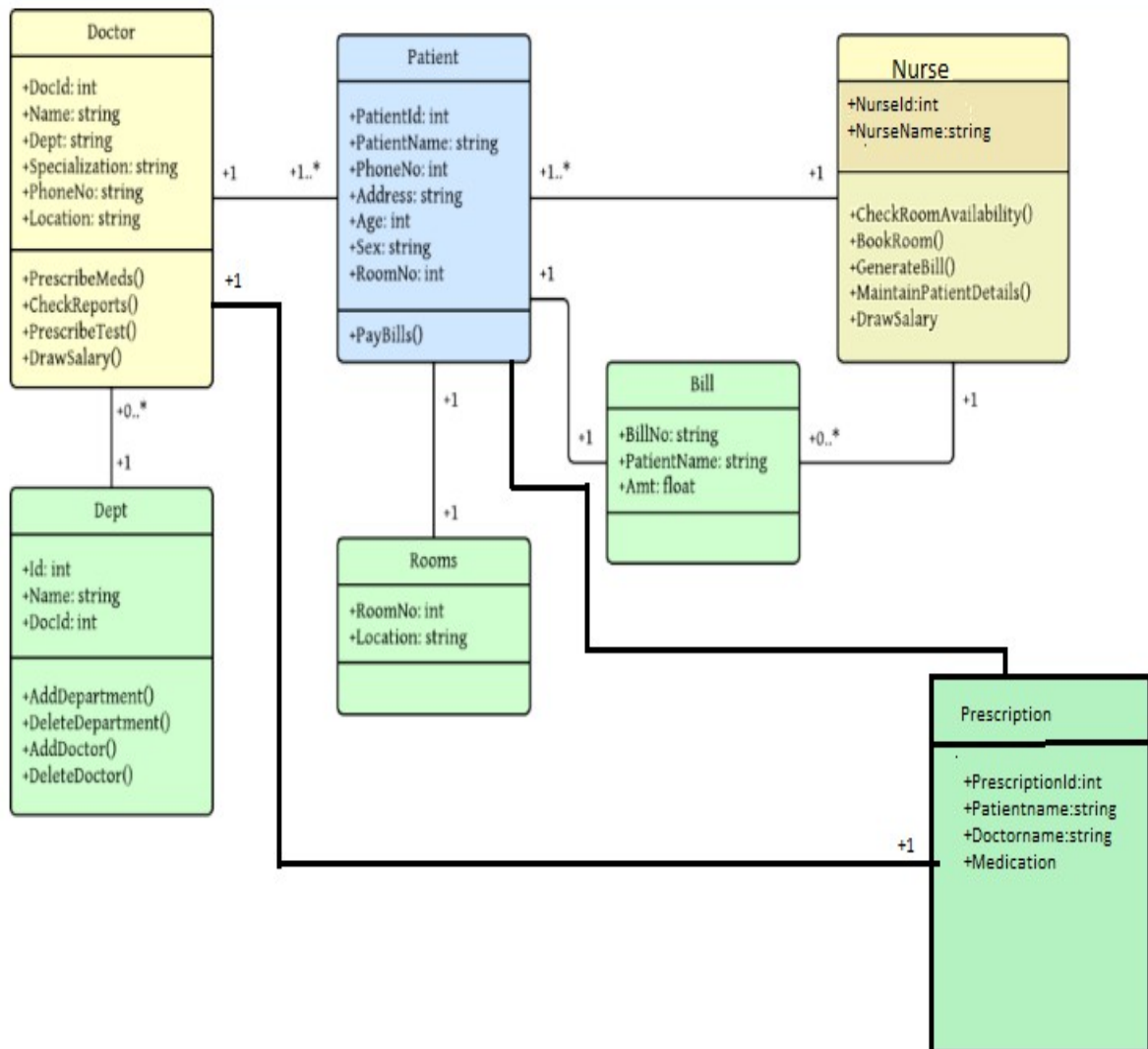
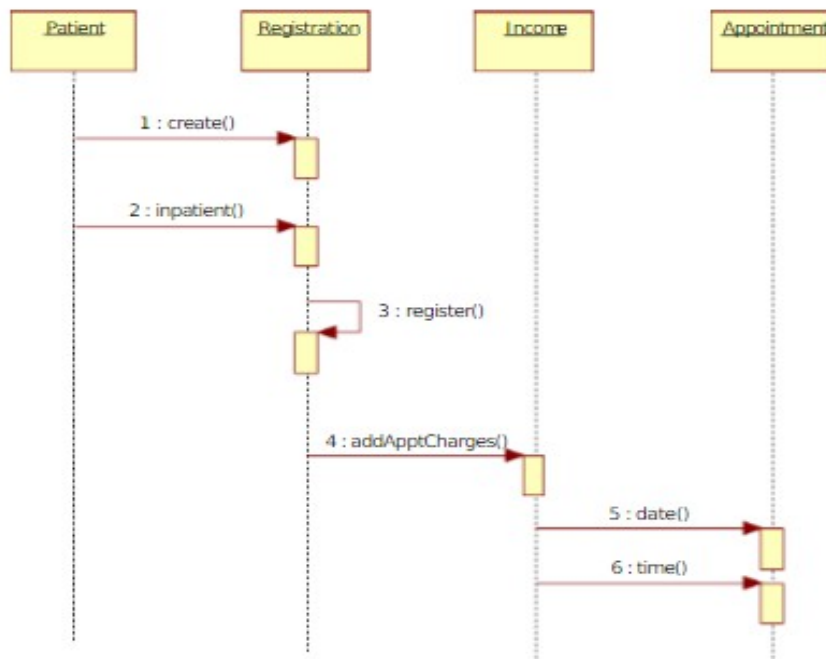


Figure:4.6 Class Diagram of Hospital Management System

## 4.3 Sequence Diagram

A Sequence diagram is a communication diagram that shows how actions work with one another and in what order. It is a concept of a Message Sequence Chart. A sequence diagram shows object interaction arranged in time sequence [16].

### 4.3.1 Sequence Diagram for Patient



Figure

e:4.7 Sequence Diagram for patient

### 4.3.2 Sequence Diagram for Doctor

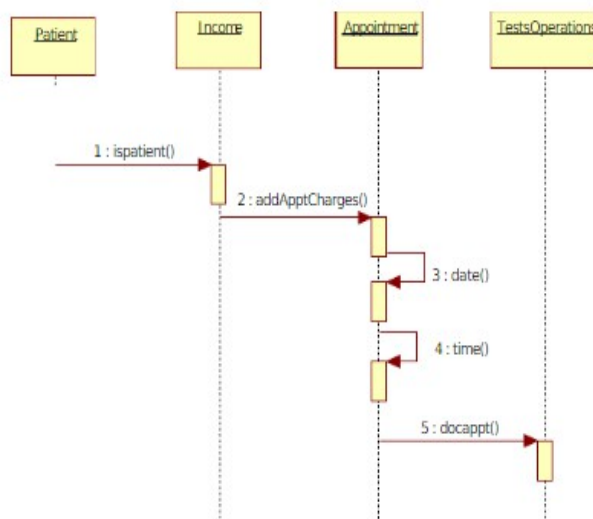


Figure: 4.8 Sequence Diagram for Doctor

### 4.3.3 Sequence Diagram of Report

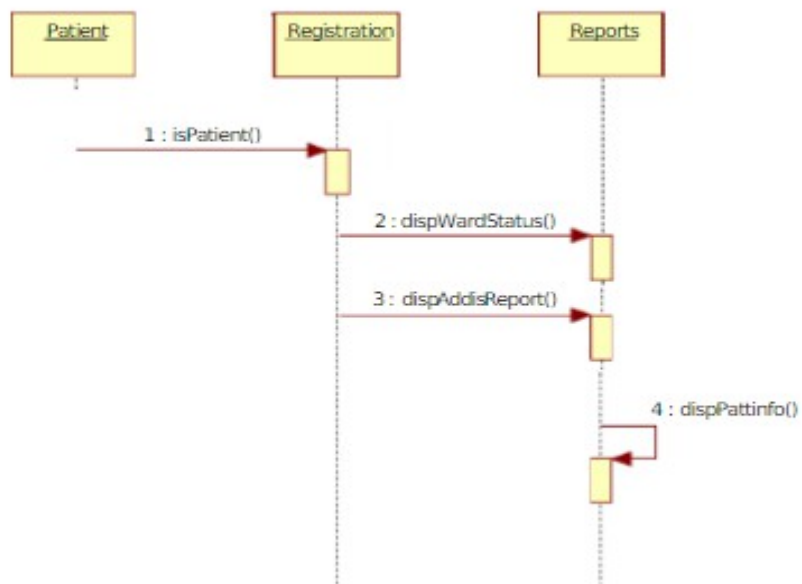


Figure: 4.9 Sequence Diagram of Report

### 4.3.4 Sequence Diagram of Hospital Management

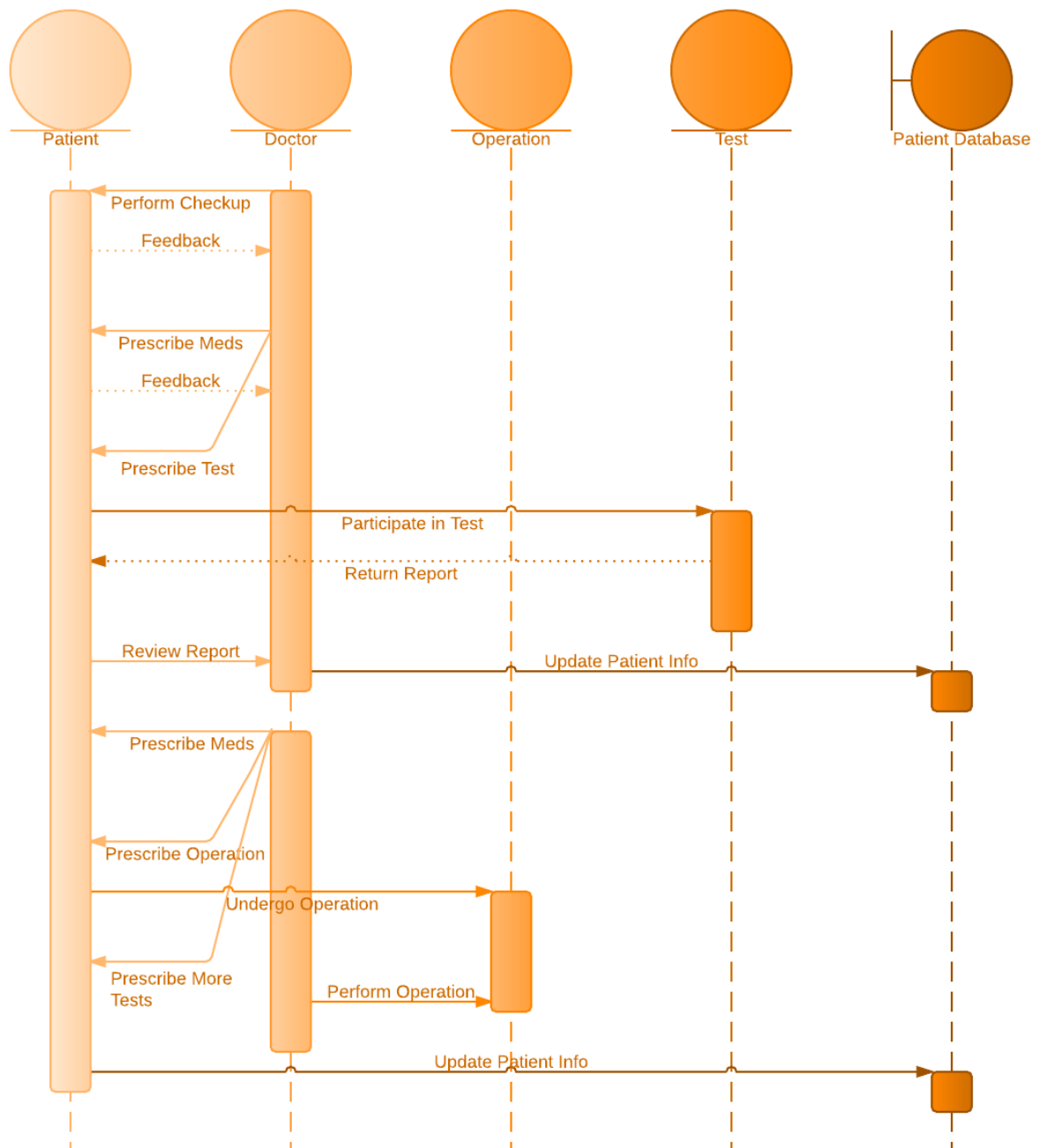


Figure: 4.10 Sequence Diagram of Hospital Management

## 4.4 Entity-Relationship Model

An entity-relationship model defines inter-related things of attention in a specific domain of information. An ER model is collected of entity types and stipulates relationships that can exist among cases of those entity types [18].

### 4.4.1 Entity-Relationship Diagram

An entity-relationship diagram (ERD) is a graphical image of an evidence scheme that displays the association among people, objects, places, ideas or events within that scheme[17]. An ERD is a data modeling method that can help express business processes and can be used as the basis for a relational database[19].

### 4.4.2 Relational Model for Hospital Management System

Administrator (id, username, email, password)

Department(id, name, description)

Laboratorist(id, name, email, password, address, phone)

Nurse(nurse\_id,name, email, password, address)

Report(id,type, description, timestamp, doctor\_id, patient\_id)

Doctor(doctor\_id,name,email, password, address, phone)

Medicine (med\_id, name, medicine\_catagory, price, manufacturing\_company, status)

Pharmacist (id, name, adderss)

Patient (id, name, address, sex, email, password, birth\_date, age, blood\_group, account\_opening timestamp)

Payment (id, type, invoice\_id, patient\_id, amount)

### 4.4.3E-R Diagram for Hospital Management System

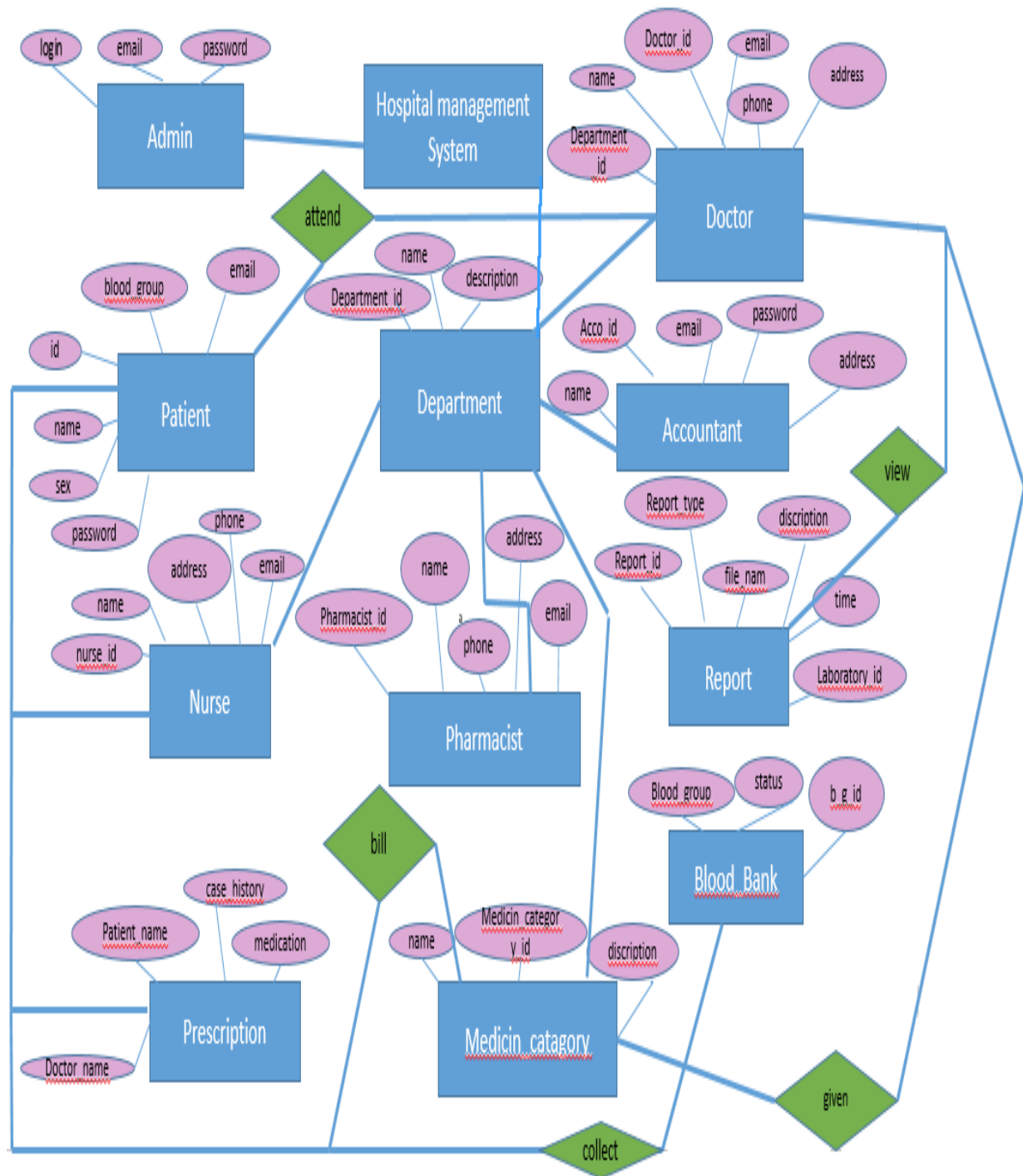


Figure 4.10 E-R Diagrams for Hospital Management System

## **CHAPTER 5**

---

### **Data Definitions and Form Design**



## 5.1 Data Definition














This section describes the tables those are used in the hospital management system.

### 5.1.1 Database hospital management


















#### Table:

Different types of information tables those are used in this system discussed.

#### 5.1.2 Table structure for admin

	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	<b>admin_id</b> 	int(11)			No	None	AUTO_INCREMENT	 Change  Drop
<input type="checkbox"/>	2	<b>name</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	3	<b>email</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	4	<b>password</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	5	<b>address</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	6	<b>phone</b>	longtext			No	None		 Change  Drop

#### 5.1.3 Table structure for doctor

	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	<b>doctor_id</b> 	int(11)			No	None	AUTO_INCREMENT	 Change  Drop
<input type="checkbox"/>	2	<b>name</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	3	<b>email</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	4	<b>password</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	5	<b>address</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	6	<b>phone</b>	longtext			No	None		 Change  Drop
<input type="checkbox"/>	7	<b>department_id</b>	int(11)			No	None		 Change  Drop
<input type="checkbox"/>	8	<b>profile</b>	longtext			No	None		 Change  Drop

## 5.1.4 Table structure for pharmacist

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>pharmacist_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>email</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>password</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 5	<b>address</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 6	<b>phone</b>	longtext			No	None		Change  Drop

## 5.1.5 Table structure for nurse

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>nurse_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>email</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>password</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 5	<b>address</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 6	<b>phone</b>	longtext			No	None		Change  Drop

## 5.1.6 Table structure for bed\_allotment

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>bed_allotment_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>bed_id</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 3	<b>patient_id</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 4	<b>allotment_timestamp</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 5	<b>discharge_timestamp</b>	int(11)			No	None		Change  Drop

## 5.1.7 Table structure for medicine\_catagory

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>medicine_category_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>description</b>	longtext			No	None		Change  Drop

## 5.1.8 Table structure for accountant

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>accountant_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>email</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>password</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 5	<b>address</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 6	<b>phone</b>	longtext			No	None		Change  Drop

## 5.1.9 Table structure for patient

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>patient_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>email</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>password</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 5	<b>address</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 6	<b>phone</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 7	<b>sex</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 8	<b>birth_date</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 9	<b>age</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 10	<b>blood_group</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 11	<b>account_opening_timestamp</b>	int(11)			No	None		Change  Drop

## 5.1.10 Table structure for diagnosis\_report

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>diagnosis_report_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>report_type</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>document_type</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>file_name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 5	<b>prescription_id</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 6	<b>description</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 7	<b>timestamp</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 8	<b>laboratorist_id</b>	int(11)			No	None		Change  Drop

### 5.1.11 Table structure for laboratorist

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>laboratorist_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>name</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>email</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>password</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 5	<b>address</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 6	<b>phone</b>	longtext			No	None		Change  Drop


















### 5.1.12 Table structure for blood\_bank

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>blood_group_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>blood_group</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>status</b>	longtext			No	None		Change  Drop

### 5.1.13 Table structure for payment

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<b>payment_id</b>	int(11)			No	None	AUTO_INCREMENT	Change  Drop
<input type="checkbox"/> 2	<b>payment_type</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 3	<b>transaction_id</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 4	<b>invoice_id</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 5	<b>patient_id</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 6	<b>method</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 7	<b>description</b>	longtext			No	None		Change  Drop
<input type="checkbox"/> 8	<b>amount</b>	int(11)			No	None		Change  Drop
<input type="checkbox"/> 9	<b>timestamp</b>	int(11)			No	None		Change  Drop

### 5.1.14 Table structure for prescription

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	prescription_id 	int(11)			No	None	AUTO_INCREMENT	 Change  Drop
<input type="checkbox"/> 2	creation_timestamp	int(11)			No	None		 Change  Drop
<input type="checkbox"/> 3	doctor_id	int(11)			No	None		 Change  Drop
<input type="checkbox"/> 4	patient_id	int(11)			No	None		 Change  Drop
<input type="checkbox"/> 5	case_history	longtext			No	None		 Change  Drop
<input type="checkbox"/> 6	medication	longtext			No	None		 Change  Drop
<input type="checkbox"/> 7	medication_from_pharmacist	longtext			No	None		 Change  Drop
<input type="checkbox"/> 8	description	longtext			No	None		 Change  Drop


## 5.2 Form Design

This section describes form those are used into the system.

### 5.2.1 Loginpage

Following picture is the login page(front page). Every members are easily login their individual page with account type, email, password.

Hospital Management System



login

account type
▼

✉

🔑

login

[forgot password?](#)

Figure: 5.1 login page

### 5.2.2 Admin page

Before sign in admin need to sign up or create account with required information. After login Admin can view this page.

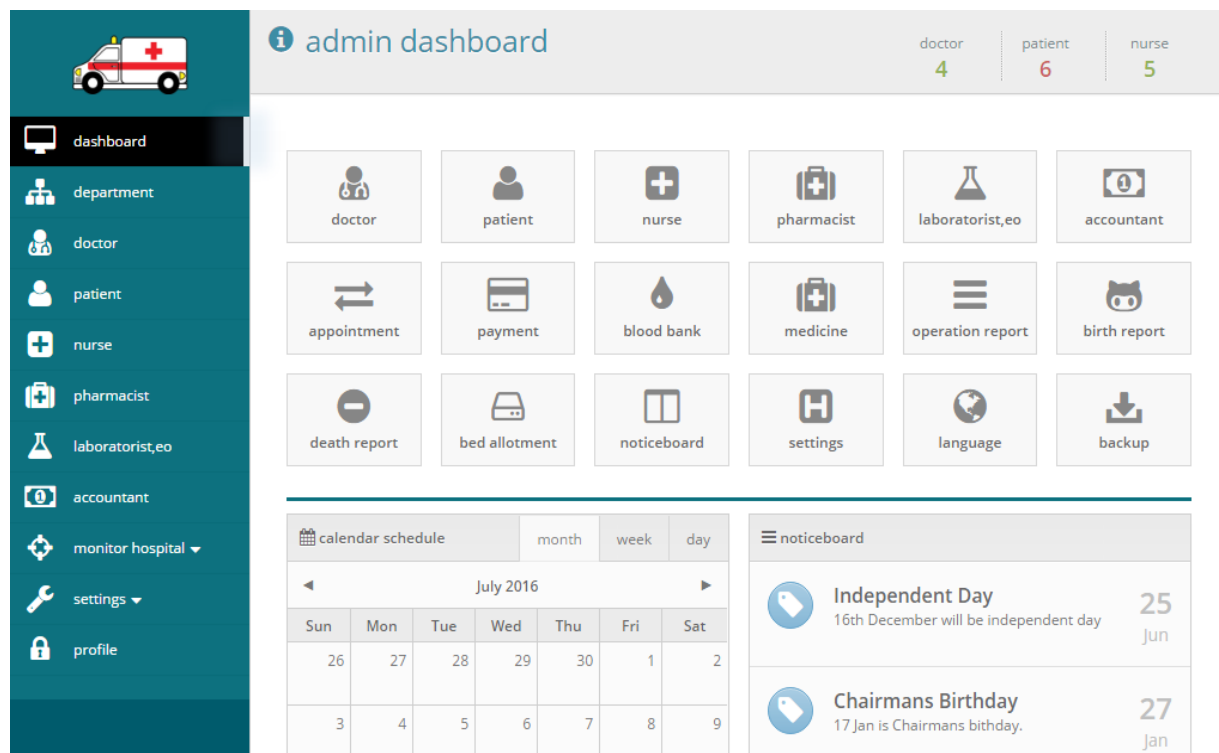


Figure: 5.2 Admin page

## 5.2.3 Medicine\_catagory page

All category of medicines and description here this page pharmacist, doctor, patients view all information.

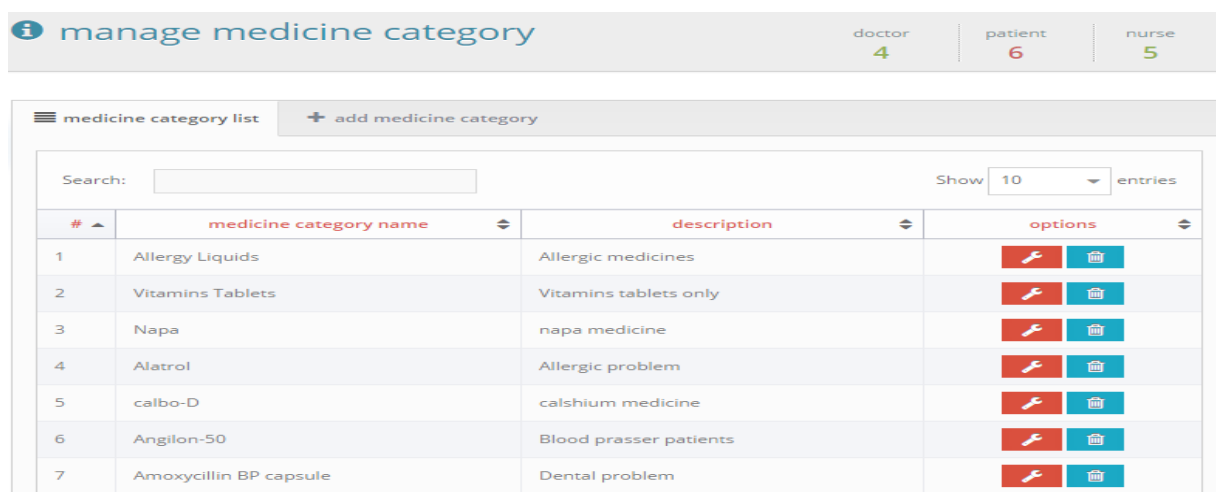


Figure: 5.3 Medicine\_catagory page

## 5.2.4 Payment page

Here is view patient's payment description. The status row show payment paid or unpaid.

manage invoice							
				doctor	patient	nurse	
				4	6	5	









  

invoice list	+ add invoice
--------------	---------------

Search:	<input type="text"/>	Show	10	entries
---------	----------------------	------	----	---------

#	invoice	title	amoun	patient	date	status	option
1	4	Daibatis test	250	Ms hasi	25 Jun,2016	paid	 
2	3	urine test	440	Mr Alamin	19 Jun,2016	paid	 
3	2	Fracture	1200	Ms Fatema	01 Dec,2015	unpaid take cash payment	 
4	1	Blood Test	500	Mr Rafiq	01 Dec,2015	paid	 

Showing 1 to 4 of 4 entries	First	Previous	1	Next	Last
-----------------------------	-------	----------	---	------	------

Figure: 5.4 payment page

## 5.2.5Dignosis\_report page

View the patients report in page. Patient name, doctor name and date include the page.

manage report					
			doctor	patient	nurse
			4	6	5




  

operation	birth	death	other	+ add report
-----------	-------	-------	-------	--------------

Search:	<input type="text"/>	Show	10	entries
---------	----------------------	------	----	---------

#	description	date	patient	doctor	options
1	apendiside	10 Jun,2016	Mr Rafiq	Dr. Abdul Karim	
2	kidney problem	24 Jun,2016	Rubana Nasrin	Dr.jafor	
3	Eye	10 Jun,2016	Mr Alamin	Dr. Abdul Karim	

Showing 1 to 3 of 3 entries	First	Previous	1	Next	Last
-----------------------------	-------	----------	---	------	------

Figure: 5.5 Dignosis\_report page

## 5.2.6 Bed\_allotment page

This page view which patient admitand what type of bed.It has show bad number,date.

manage bed allotment				doctor 4	patient 6	nurse 5
bed allotment list + add bed allotment						
Search: <input type="text"/>				Show	10	entries
#	bed number	bed type	patient	allotment date time	discharge date time	options
1	W1	ward	Ms hasi	10 Dec,2015	12 Dec,2015	
2	W2	ward	Mr Rafiq	03 Jun,2016	04 Jun,2016	
3	I1	icu	Mr kamal	14 Jun,2016	16 Jun,2016	
4	W3	ward	Mr Alamin	11 Jun,2016	10 Jun,2016	
5	W5	cabin	Ms Fatema	05 Jul,2016	09 Jul,2016	
Showing 1 to 5 of 5 entries				First	Previous	1 Next Last

Figure: 5.6 Bed\_allotment page

## 5.2.7 Add patient page

The patients' information can find in this section. Admin can add patient, delet etc.

Hospital Management System				doctor panel	select language	account
manage patient				doctor 4	patient 6	nurse 5
patient list + add patient						
Search: <input type="text"/>				Show	10	entries
#	patient name	age	sex	blood group	birth date	options
1	Ms Fatema	34	female	O+	2016-06-10	
2	Mr kamal	56	male	A+	2016-06-04	
3	Mr Rafiq	45	male	A+	2016-06-01	
4	Mr Alamin	37	male	AB+	2016-06-06	
5	Ms hasi	24	female	B+	2016-06-13	
6	Rubana Nasrin	20	female	O+	3/12/1987	
Showing 1 to 6 of 6 entries				First	Previous	1 Next Last

Figure: 5.7 add patient page

## 5.2.8 Blood\_donor page

This page view blood\_donor information that name,age,sex,blood\_group,donation date etc.



*i* manage blood donor

doctor

4

patient

6

nurse

5

blood donor list

+ add blood donor

Search:

Show 10 entries

#	name	age	sex	blood group	last donation date	options
1	Sudhir Thakur	25	male	A+	14 Oct,2014	
2	zafor hosian	36	male	AB+	20 Jun,2016	
3	Md Azizur rahman	30	male	O+	11 May,2016	
4	Rupa hasna	27	female	B+	08 Jul,2016	
5	Pushpendu Dus	34	male	AB+	06 Apr,2015	

Showing 1 to 5 of 5 entries

First

Previous

1

Next

Last

Figure: 5.8 blood\_donor page

## 5.2.9Blood\_bank page

The blood\_bank page has doner name, age, sex, blood\_group, last donate date etc.

*i* view blood bank

doctor

4

patient

6

nurse

5

blood donor list

blood bank

Search:

Show 10 entries

#	name	age	sex	blood group	last donation date
1	Sudhir Thakur	25	male	A+	14 Oct,2014
2	zafor hosian	36	male	AB+	20 Jun,2016
3	Md Azizur rahman	30	male	O+	11 May,2016
4	Rupa hasna	27	female	B+	08 Jul,2016
5	Pushpendu Dus	34	male	AB+	06 Apr,2015

Showing 1 to 5 of 5 entries

First

Previous

1

Next

Last

Figure: 5.9 blood\_bank page

## 5.2.10ViewDoctorpage

Doctor view appointment list, patient prescription, manage report etc.

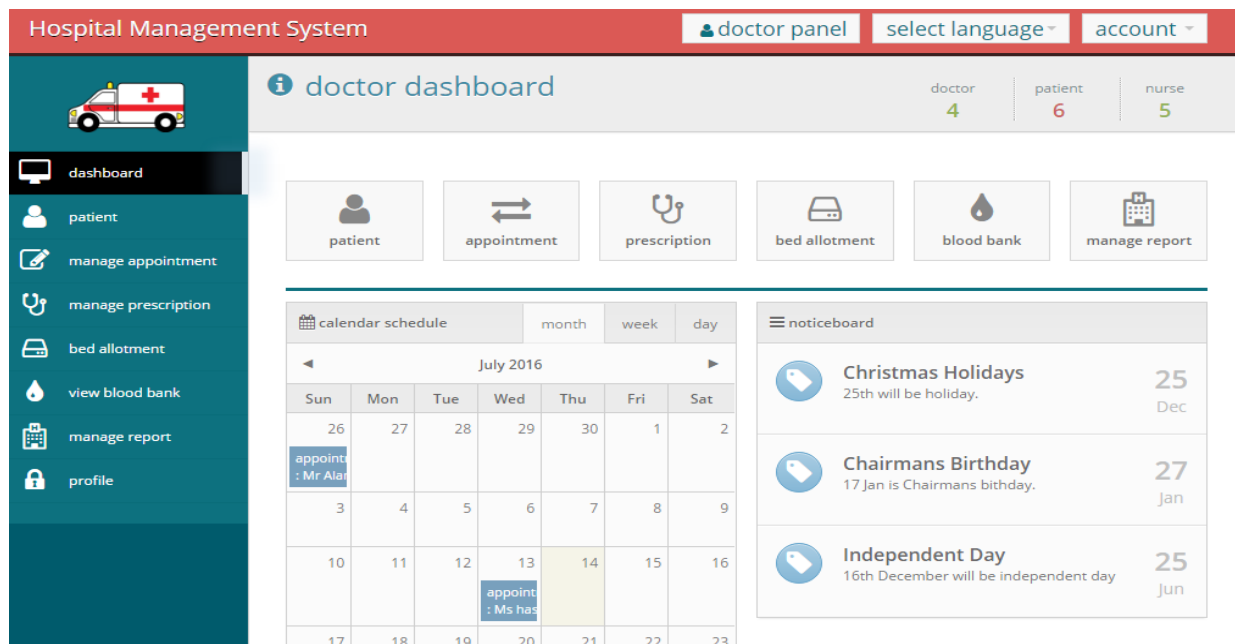


Figure: 5.10 doctor page

## 5.2.11 Pharmacist page

Pharmacist page include madicinecategory,manage medicine and provide medication.

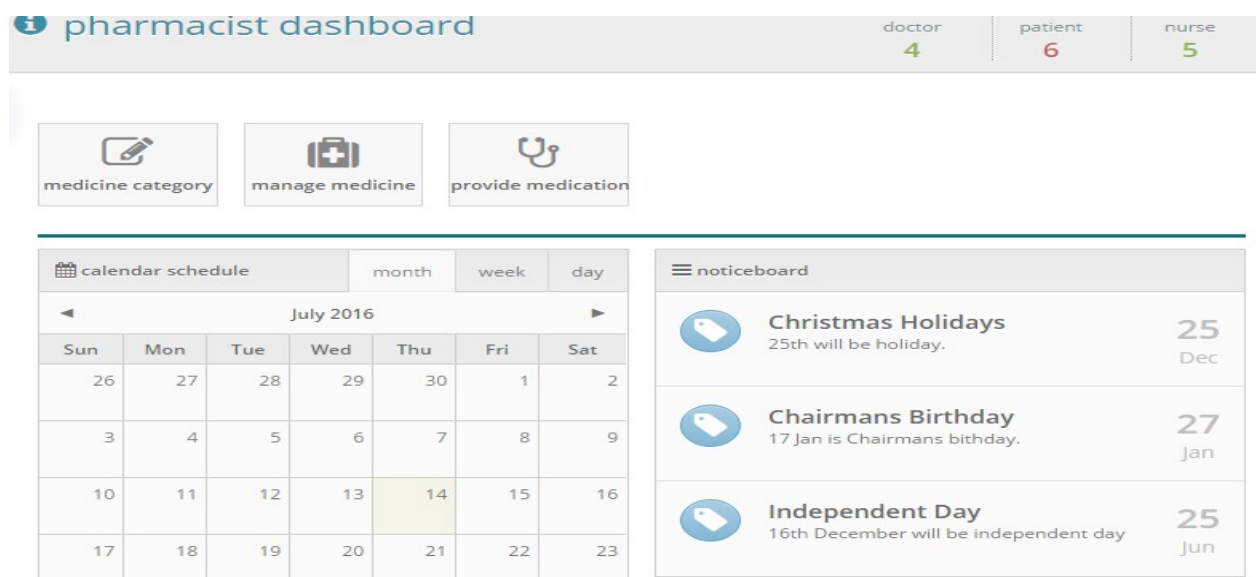


Figure: 5.11 pharmacist page

## 5.2.12Add prescription page

This page view prescription list of the patient name, doctor name, date.

prescription list

+

add prescription

Search:

Show

10

entries

# ▲	date ▼	patient ▼	doctor ▼	options ▼
1	01 Dec,2015	Ms hasi	Dr. Harun-ur rashid	<div></div> <div></div>
2	07 Jan,2016	Mr Rafiq	Dr. Harun-ur rashid	<div></div> <div></div>
3	12 Jun,2016	Ms Fatema	Dr. Abdul Karim	<div></div> <div></div>
4	24 Jun,2016	Mr Alamin	Dr.jafor	<div></div> <div></div>

Showing 1 to 4 of 4 entries

First

Previous

1

Next

Last

Figure: 5.12 add prescription page

# **CHAPTER 6**

---

## **MAINTENANCE**

### **6.1 Software Maintenance**

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes. A public awareness of maintenance is that it just includes setting defects. However, a study specified that more than 80% of maintenance is used for non-corrective movements [20]. This insight is continued by users acquiescing problem reports that in certainty are functionality enrichments to the system [21].

The maintenance process model described in IEEE, the Standard for Software Maintenance, starts the software maintenance effort during the post-delivery stage and discusses items such as planning for maintenance and measures outside the process model[21]. That process model with the IEEE maintenance phases is depicted in Figure .

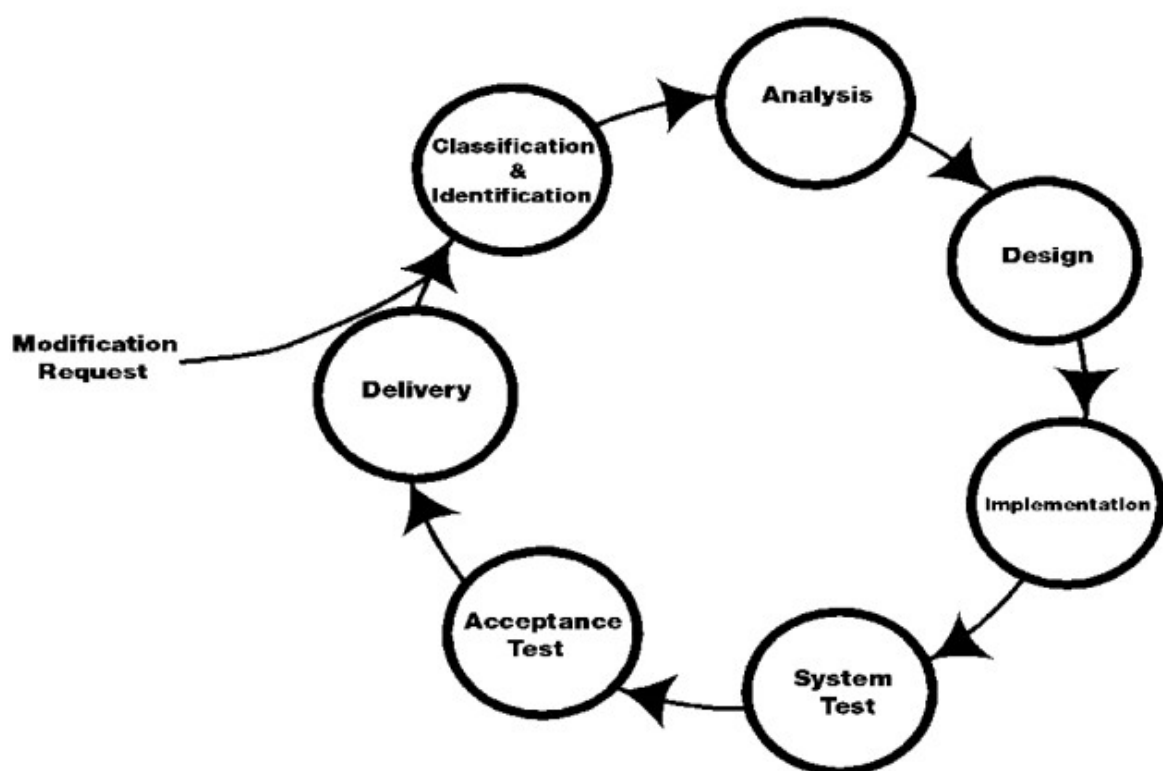


Figure : 6.1 Maintenance process model

### 6.1.1 Maintenance Activities

In the late 1970s, a renowned and extensively cited review study by Lientz and Swanson showing the actual great portion of life-cycle costs that were being used on maintenance. They considered maintenance activities into four classes [22]

- Adaptive Maintenance
- Corrective Maintenance
- Perfective Maintenance
- Preventive Maintenance

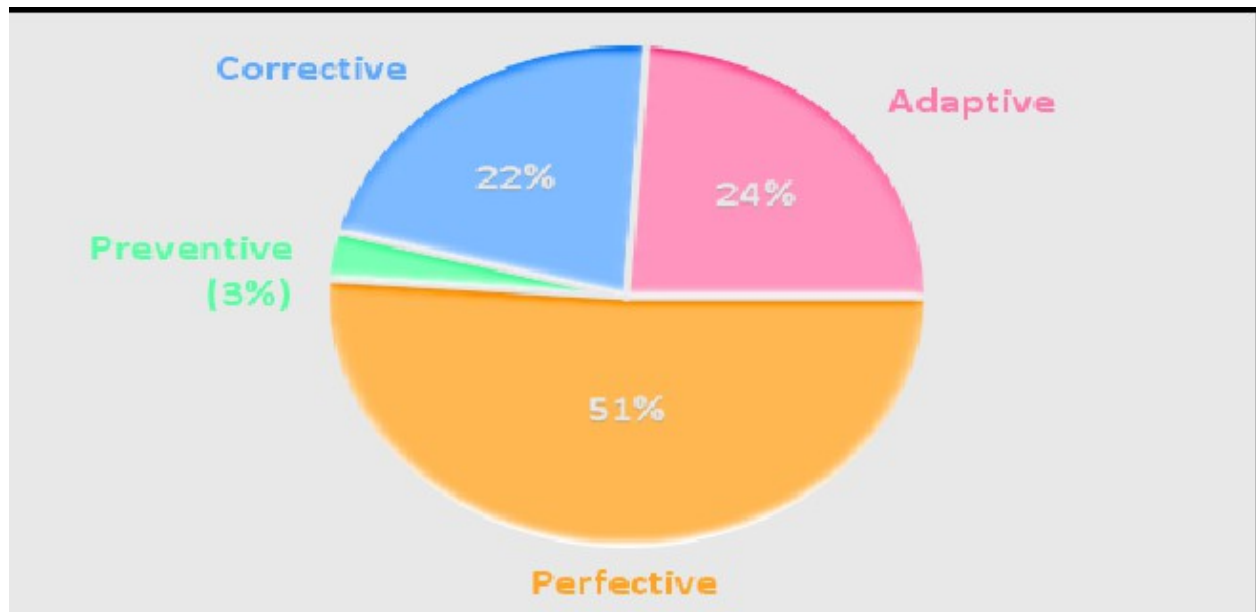


Figure: 6.2 Maintenance Activities

Adaptive Maintenance is practical when alterations in the outside situation precipitate changes to software. It contracts with adjusting the software to new surroundings.

Corrective Maintenance does to precise mistakes that are exposed after the software is in practice. It contracts with fitting bugs in the code.

Perfective Maintenance includes enrichments that are wished by user communal. It contracts with appraising the software permitting to changes in user necessities.

Preventive Maintenance advances upcoming maintainability and dependability and delivers a foundation for future enrichment. It contracts with informing certification and constructing the software more maintainable.

## 6.1.2 Difficulties of Maintenance

There are four key difficulties that can sluggish down the maintenance process.

- Formless code
- Maintenance programmers having inadequate knowledge of the system
- Documentation being inattentive
- Out of Date or inadequate

The achievement of the maintenance stage trusts on these problems being earlier in the life sequence. In our system we try to follow the maintenance phases to complete those activities. The code is structured and we have adequate knowledge about programming [22].

## **CHAPTER 7**

---

### **CONCLUSION AND FUTURE WORK**

#### **7.1 Conclusion**



The Project Hospital Management System (HMS) is for computerizing the working in a hospital. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The hospital managing system was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital. It generates test reports and also provides the facility for searching the details of the patients. It also billing facility on the basis of patient's status whether it is an indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

## **7.2 Limitations**

For upcoming improvement, there are some proposals to advance our project abilities.

- There is no email verification or confirmation system.
- Does not upload notice timely.
- There is no online bill payment system.
- Can't upload more than 10 Mega Bytes file.

## **7.3 Future Plans**

- Will add more features to improve my project.
- There will be email verification system.
- Provide health tips for patient.
- SMS alert system is easier for the patients.
- I hope also work on online bill payment system.
- Additionally, it is just a beginning. Supplementary the system may be used in various other types of reviewing process

## **REFERENCES**

- [1]. [Online]. Available: <http://www.Hospital Management and Information System>. International Journal of Sciences: Basic and Applied Research (IJSBAR) ISSN 23074531.
- [2]. [Online]. Available: Pitre,R An investigation of the relationship between hospital management information system and hospital effectiveness.
- [3]. [Online]. Available: <http://www.rss.jo/default/file Hospital Management System.pdf>.
- [4]. [Online]. Available: "What are the Software Development Models?", Istqbexamcertification.com, Available: <http://istqbexamcertification.com/what-are-the-software-development-models/>.
- [5]. [Online]. Available: "Waterfall model ",Wikipedia. <http://en.Wikipedia.org/Wiki/Waterfall model>.
- [6]. [Online]. Available: "What is Waterfall model- advantages, disadvantages and when to use it?",Istqbexamcertification.com.2016.[Online].Available: <http://istqbexamcertification.com/what-is-waterfall-model-advantagesdisadvantages-and-when-to-use-it/>.
- [7]. [Online]. Available: "SDLC - Waterfall Model", www.tutorialspoint.com.Available: [http://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm).
- [8]. [Online]. Available: "What are the Software Development Life Cycle (SDLC) phases?", Istqbexamcertification.com, 2016. [Online]. Available: <http://istqbexamcertification.com/what-are-the-software-development-life-cycle-sdlc-phases/>.
- [9]. [Online]. Available: "Requirements analysis", Wikipedia, 2016. Available: [https://en.wikipedia.org/wiki/Requirements\\_analysis](https://en.wikipedia.org/wiki/Requirements_analysis).
- [10]. [Online]. Available: "Functional requirement", Wikipedia. Available: [https://en.wikipedia.org/wiki/Functional\\_requirement](https://en.wikipedia.org/wiki/Functional_requirement).
- [11]. [Online]. Available: <http://> L. Chen, M. Ali Babar and B. Nuseibeh, "Characterizing Architecturally Significant Requirements", IEEE Software, vol. 30, no. 2, pp. 38-45, 2013.
- [12]. [Online].Available:"SOFTWARE REQUIREMENTS SPECIFICATION", Academia.edu, [Online].Available:[http://www.academia.edu/15064459/SOFTWARE\\_REQUIREES SPECIFICATION](http://www.academia.edu/15064459/SOFTWARE_REQUIREES_SPECIFICATION)

- [13]. [Online]. Available: "Use case", Wikipedia, 2016. Available: [https://en.wikipedia.org/wiki/Use\\_case](https://en.wikipedia.org/wiki/Use_case)
- [14]. [Online]. Available: "Actor (UML)", Wikipedia, 2016. Available: [https://en.wikipedia.org/wiki/Actor\\_\(UML\)](https://en.wikipedia.org/wiki/Actor_(UML)).
- [15]. [Online]. Available: <http://csis.pace.edu/~marchese/CS389/L9/Use%20Case%20Diagrams.pdf>.
- [16]. [Online]. Available: "Sequence diagram", Wikipedia, Available: [http://en.Wikipedia.org/Wiki/Sequence\\_diagram](http://en.Wikipedia.org/Wiki/Sequence_diagram).
- [17]. [Online]. Available: "UML - Activity Diagrams", [www.tutorialspoint.com](http://www.tutorialspoint.com), Available: [http://www.tutorialspoint.com/uml/uml\\_activity\\_diagram.htm](http://www.tutorialspoint.com/uml/uml_activity_diagram.htm).
- [18]. [Online]. Available: <http://> P. Chen, "The entity-relationship model---toward a unified view of data", ACM Transactions on Database Systems, vol. 1, no. 1, pp. 9-36, 1976.
- [19]. [Online]. Available: "What is entity relationship diagram? - Definition from WhatIs.com", Search CRM, 2016. [Online]. Available: <http://searchcrm.techtarget.com/definition/entity-relationship-diagram>.
- [20]. [Online]. Available: <http://> Pigoski, Thomas M., 1997: Practical software maintenance: Best practices for managing your software investment.
- [21]. [Online]. Available: <http://> S. Eick, T. Graves, A. Karr, J. Marron and A. Mockus, "Does code decay? Assessing the evidence from change management data", IEEE Transactions on Software Engineering, vol. 27, no. 1, pp. 1-12, 2001.
- [22]. [Online]. Available: "Software maintenance", Wikipedia, Available: [https://en.wikipedia.org/wiki/Software\\_maintenance](https://en.wikipedia.org/wiki/Software_maintenance).