

## **Abstract**

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

The project 'Hospital Management System' is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses HTML and CSS as the front-end software's and has connectivity with MY SQL and PHP.

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Medicine details, Doctor, Wards, Admin, Store, bill payment, record modification, discharge details etc.

## Table of Contents

Abstract	1
List of figures	2
<b>Chapter 1</b>	
Introduction	6
1.1 Problem Statement	6
1.2 Objective	7
1.3 Scope	7
<b>Chapter 2</b>	
Project Management	8
2.1 Project planning and scheduling	8
2.1.1 Methodology	8
2.1.2 Project Management Life Cycle	9
2.2 Risk Management	10
<b>Chapter 3</b>	
System Analysis	11
3.1 Background Study	11
3.2 Software system attributes	11
3.2.1 Reliability	11
3.2.2 Availability	11
3.3.3 Security	11
3.3 Scope of working	12
3.4 Feasibility study	13
3.4.1 Technical Feasibility	13
3.4.2 Operational Feasibility	13
3.4.3 Economic Feasibility	13
3.4.4 Management Feasibility	13
3.4.5 Social Feasibility	13

## **Chapter 4**

System Design	14
4.1 Database Design	14
4.2 E-R Diagram of Hospital Management System	15
4.3 Database schema of Hospital Management System	16
4.4 Data Flow Diagram of Hospital Management System	17
4.5 User Interface	18
4.5.1 Home Page	19
4.5.2 Admin Login page	20
4.5.3 Admin dashboard page	21
4.5.4 Patient details page	22
4.5.5 Employee details page	23
4.5.6 Pharmacy details page	24.
4.5.7 Accounting page	25
4.5.8 Inventory page	26
4.5.9 Reporting page	27
4.5.10 Medical Record page	28
4.5.11 Laboratory details page	29
4.5.12 Surgical/Theatre details page	30
4.5.13 Payroll management page	31
4.5.14 Vendor management page	32
4.5.15 Doctor's login page	33
4.5.16 Doctor's dashboard page	34

## **Chapter 5**

System Implementation	35
5.1 Implementation	35
5.2 Implementation Environment	35
5.3 Functional Requirement	36
5.3.1 Administrator Interface	36
5.3.2 Doctor's Interface	36

5.3.3 Login information	36
5.3.4 Available Features	37
5.3.5 Programs used	38
5.3.6 Source codes	39
<b>Chapter 6</b>	
System testing	46
6.1 Integration Testing	46
6.2 Unit Testing	46
6.3 System Testing	46
6.4 Acceptance Testing	46
6.5 Recovery Testing	46
6.6 Functional Testing	47
6.7 Hardware/Software Testing	47
6.8 Security Testing	47
6.9 Advantages	47
<b>Chapter 7</b>	
Conclusion	48
7.1 Conclusions	48
7.2 Limitations of the system	48
<b>REFERENCES</b>	49



## List of Figures

<b>Fig.No.</b>	<b>Figure Name</b>	<b>Page No.</b>
1.	Waterfall model	8
2.	E-R Diagram of Hospital Management System	15
3.	Database schema of Hospital Management System	17
4.	Data flow diagram of Hospital Management System	18
5.	Home page	19
6.	Administrator login page	20
7.	Admin dashboard page	21
8.	Patient details page	22
9.	Employee details page	23
10.	Pharmacy details page	24
11.	Accounting page	25
12.	Inventory page	26
13.	Reporting page	27
14.	Medical Record page	28
15.	Laboratory details page	29
16.	Surgical/Theatre details page	30
17.	Payroll management page	31
18.	Vendor management page	32
19.	Doctor's login page	33
20.	Doctor's dashboard page	34

## **Chapter one**

### **1. Introduction**

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

#### **1.1 Problem Statement**

Since Hospital is associated with the lives of common people and their day-to-day routines so we decided to work on this project.

The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like Room activities, admission of new patient, discharge of patient, assign a doctor, and finally compute the bill etc. We have tried my best to make the complicated process Hospital Management System as simple as possible using Structured & Modular technique & Menu oriented interface. We have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though we cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is performing each Hospital's activity in computerized way rather than manually which is time consuming.

We are confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.





## **1.2 Objective**

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

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process. Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper. Thus, keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “Administration support system for medical institutions”.

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

## **1.3 Scope**

The proposed software product is the Hospital Management system (HMS). The system will be used in any hospital, clinic, dispensary or pathology labs. Clinic, dispensary or pathology to get the information from the patients and then storing that data for future usages. The current system in use is a paper-based system. It is too slow and cannot provide updated lists of patients within reasonable timeframe. The intention of the system is to reduce over-time pay and increase the number of patients that can be treated accurately. Requirement statements in these documents are both functional and non-functional.

## Chapter Two

### 2. Project Management

#### 2.1 Project planning and scheduling

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path.

##### 2.1.1 Methodology

We have used Iterative and Incremental Development model (IID) for our project development. This development approach is also referred to as Iterative Waterfall Development approach. Iterative and Incremental Development is a software development process developed in response to the more traditional waterfall model. This model is designed to take care of such big project. The large and complicate project chiefly demand better development and testing procedure. The waterfall model is well known for its repeated testing process. Hence, I choose the waterfall model for developing my software.

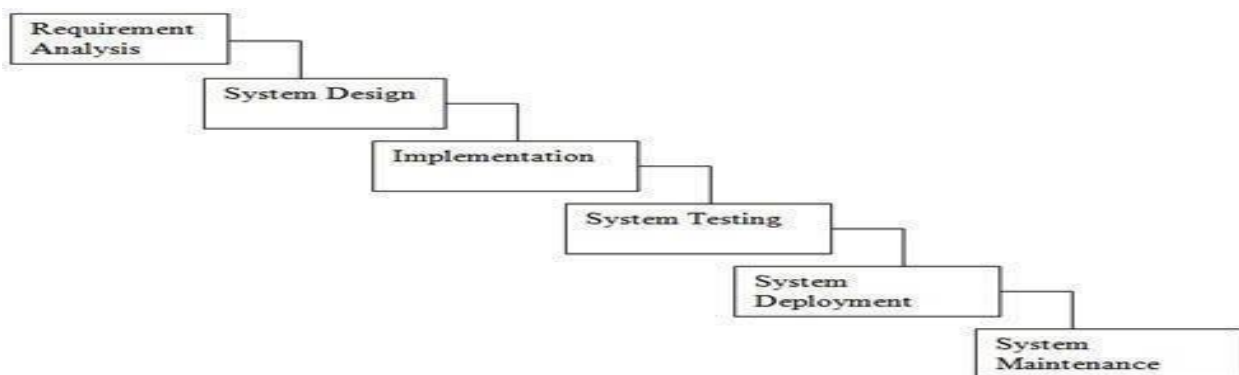


Fig. 2.1: Waterfall model

Some advantages of waterfall model:

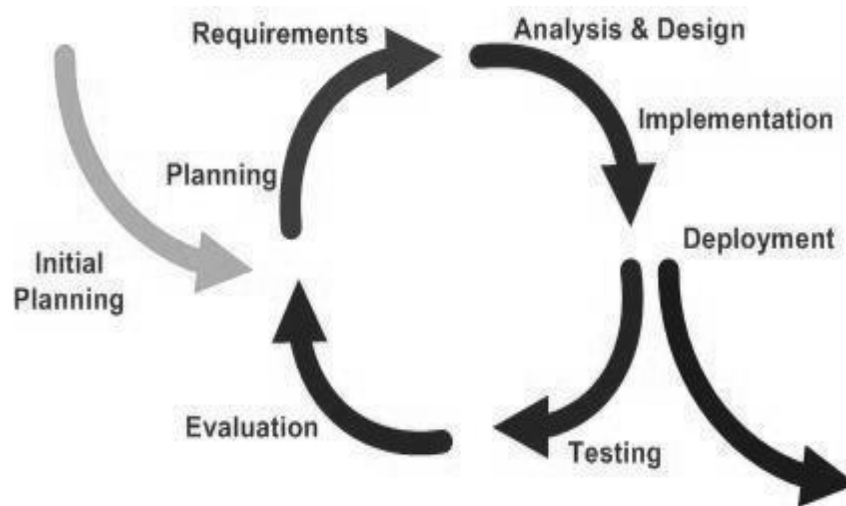
- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model.
- Phases are processed and completed one at a time
- Works well for smaller projects where requirements are very well understood.

### 2.1.2 Project Management Life Cycle

The Project Management Life Cycle has four phases. Each project life cycle phase is described along with the tasks need to complete it

The four phases is

1. Initiation
2. Planning
3. Execution
4. Closure.



**Fig. 2.2:** Iterative and Life Cycle

## **2.2 Risk Management**

Software Risk Management is a proactive approach for minimizing the uncertainty and potential loss associated with a project. Some categories of risk include product size, business impact, customer-related, process, technology, development environment, staffing (size and experience), schedule, and cost. Risk Management is a practice with processes, methods, and tools for managing risks in a project.

Risk identification is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, we can take a first step toward avoiding them when possible and controlling them when necessary. To perform the risk identification, we categorized the risk into different categories as:

1. Project Risk
2. Technical Risk
3. Business Risk
4. Known Risk
5. Predictable Risk
6. Unpredictable

## Chapter Three

### 3. System Analysis

#### 3.1 Background Study

System Analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system, it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system. One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

#### 3.2 Software system attributes

**3.1.1 Reliability:** This application is a reliable product that produces fast & verified output of all its process.

**3.1.2 Availability:** This application will be available to use and help them to carry their operations conveniently.

**3.1.3 Security:** This application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

### **3.3 Scope of working**

The proposed software product is the Hospital Management system (HMS). The system will be used in any hospital, clinic, dispensary or pathology labs. Clinic, dispensary or pathology to get the information from the patients and then storing that data for future usages. The current system in use is a paper-based system. It is too slow and cannot provide updated lists of patients within reasonable timeframe. The intention of the system is to reduce over-time pay and increase the number of patients that can be treated accurately. Requirement statements in these documents are both functional and non-functional

### 3.4 Feasibility study

**3.4.1 Technical Feasibility:** This is concerned with specifying equipment and software that will successfully satisfy the user requirement; the technical needs of the system may vary considerably, but might include:

The facility to produce outputs in a given time:

1. Response time under conditions.
2. Ability to process a certain volume of transaction at a particular seep.
3. Facility to communicate data to distant location.

**3.4.2 Operational Feasibility:** It is mainly related to human organization and political aspects. The points to be considered are:

4. What changes will be brought with the system?
5. What organizational structures are distributed?
6. What new skills will be required? Do the existing staff members have these skills? If not, can then the trained due course of time

**3.4.3 Economic Feasibility:** Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More frequently known as cost/benefit system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system.

**3.4.4 Management Feasibility:** It is a determination of whether a proposed project will be acceptable to management. If does not accept a project of gives a negligible support to it; the analyst will tend to view the project as a no feasible one.

**3.4.5 Social Feasibility:** Social feasibility is a determination of whether the project will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

## **Chapter Four**

### **4. System Design**

#### **4.1 Database Design**

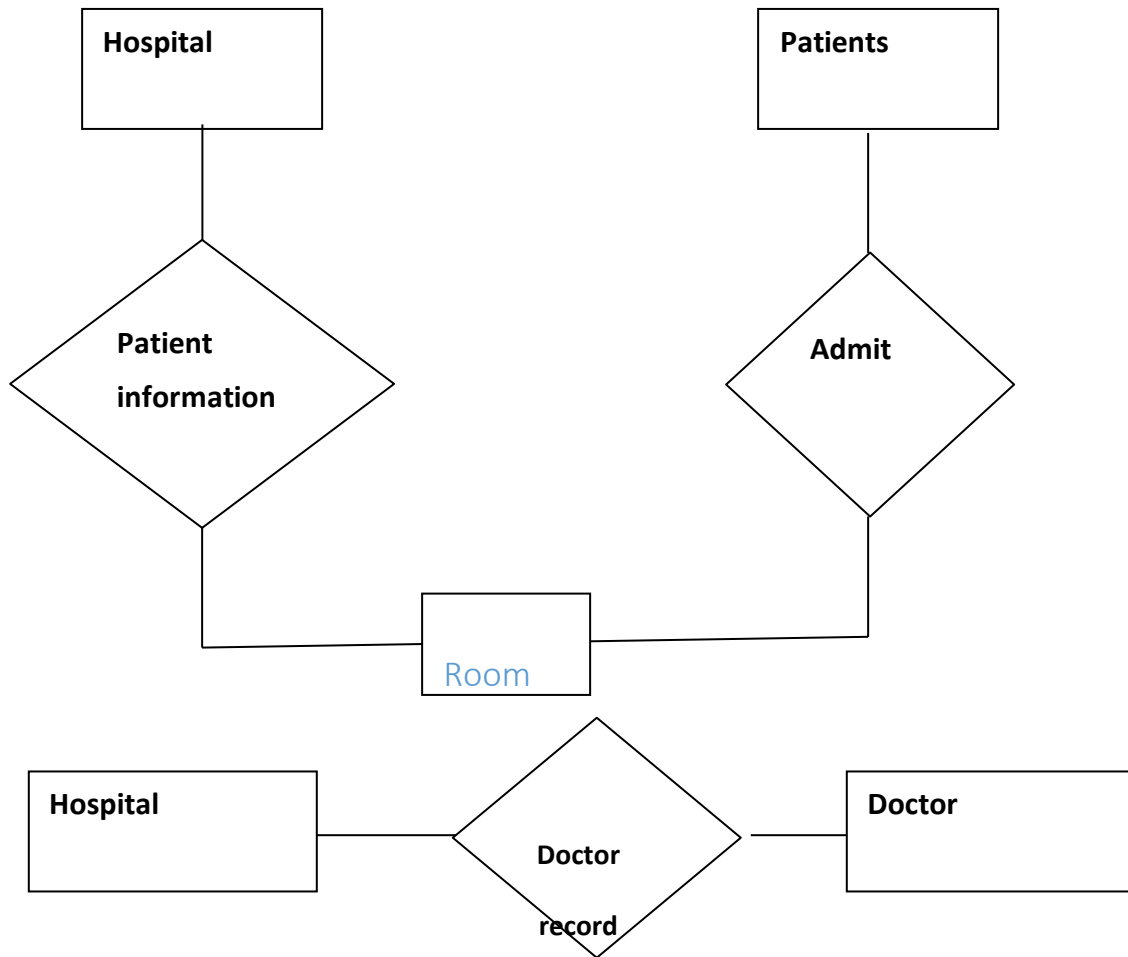
Database design is the process of producing a detailed data model of database. This data model contains all the need logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different part of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system.

#### **4.2 E-R Diagram of Hospital Management System**

An entity-relationship diagram (ERD) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.





**Fig. 4.1:** E-R Diagram of Hospital Management System

### 4.3 Database schema of Hospital Management System

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema can be divided broadly into two categories –

**Physical Database Schema:** This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

**Logical Database Schema:** This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

#### List of tables:

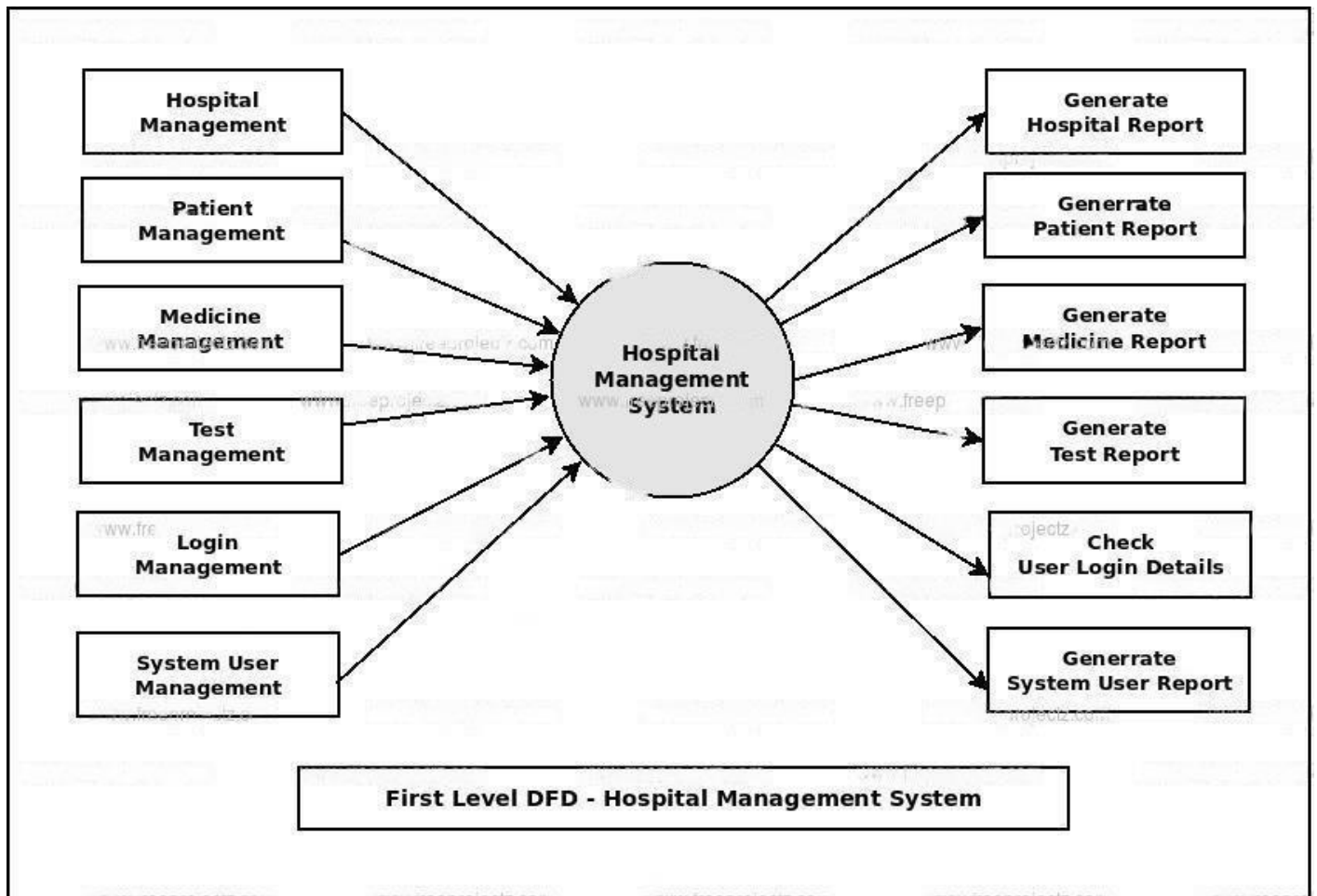
1. admin
2. Users
3. Patients
4. Physician
5. Services
6. Transactions
7. user details
8. Room
9. Discounts
10. Doctors
11. Doctor specialization.

Table	Action	Rows	Type	Collation	Size	Overhead
discounts	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 K18	-
orders	★ Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 K18	-
orders_discount	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 K18	-
orders_rooms	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 K18	-
patients	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16 K18	-
physician	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 K18	-
rooms	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 K18	-
services	★ Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	16 K18	-
transactions	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 K18	-
users	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 K18	-
visitors	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K18	-
visitor_acc	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 K18	-
13 tables	Sum	76	InnoDB	latin1_swedish_ci	103 K18	0.0

**Fig.4.2:** Database schema of Online Marketplace

#### 4.4 Data Flow Diagram of Hospital Management System

The context diagram is the most abstract data flow representation of a system. It represents the entire system as a single bubble and. The various external entities with which the system interacts and the data flows occurring between the system and the external entities are also represented. The name context diagram is well justified because it represents the context in which the system is to exist i.e. the external entities (users) that would interact with the system and specific data items they would be receiving from the system.



**Fig. 4.3:** Data flow diagram of Hospital management system

## 4.5 User Interface

### 4.5.1 Home Page

**Actor:** Any users



**Fig. 4.4:** Home page

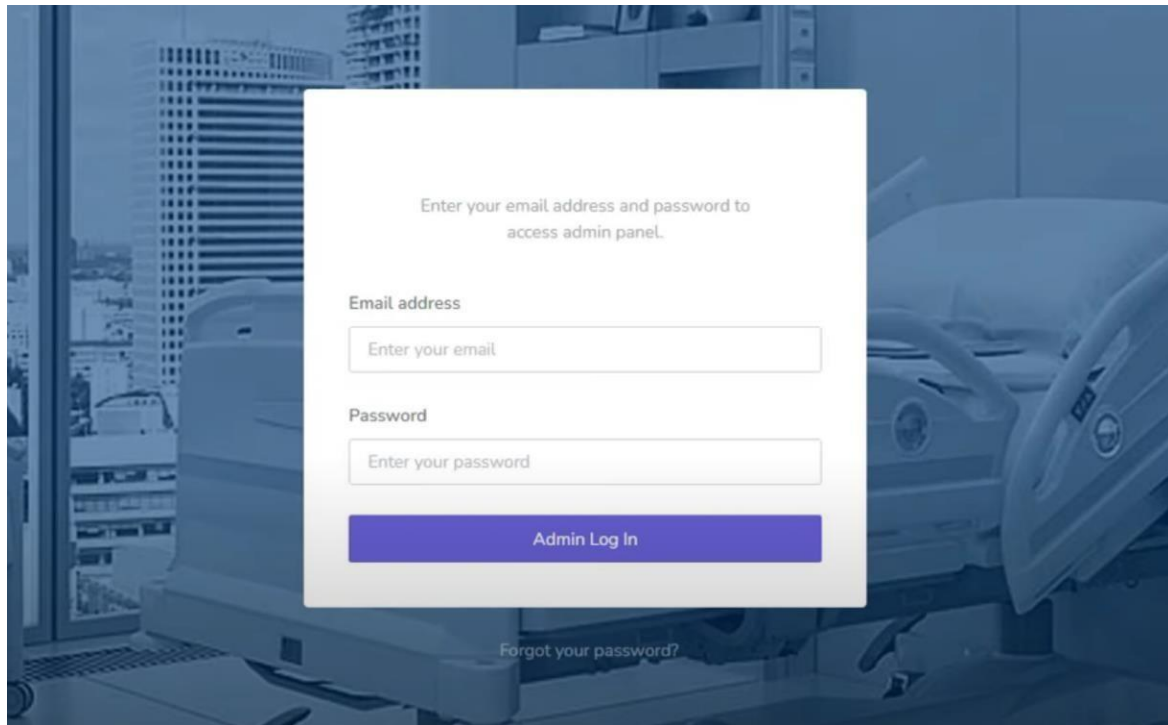
Flow:

1. Only doctors will be able access this page.

## 4.5.2 Admin login page

**Actor:** Admin users

**Output:** Hospital management system login



**Fig. 4.4:** Admin Login page

Flow:

1. Only Admin user can browsers this page.
2. Admin user can search all patient appointment and all users' activities.

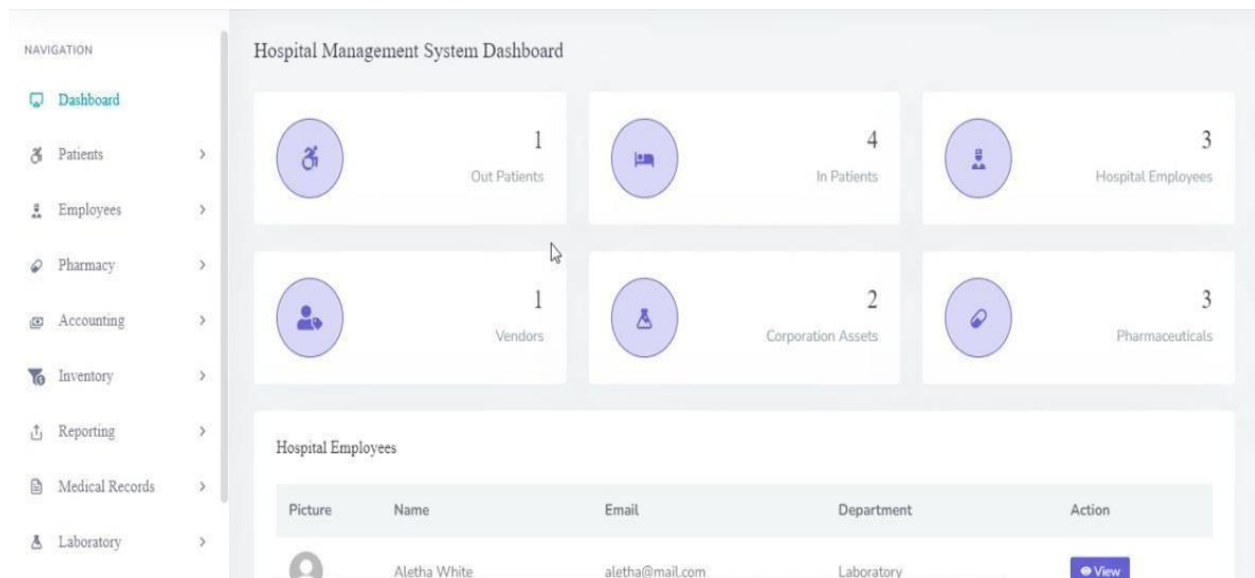
Alternate Flow:

- (1) If the user email is wrong then it is asked to login again.
- (2) If the password is wrong then the user is asked to enter again.

### 4.5.3 Admin dashboard page

**Actor:** Admin users

**Output:** Admin dashboard page



**Fig. 4.4:** Admin details page

Flow:

1. Admin user can browsers this page.
2. Admin user can view all module here

## 4.5.4 Patients details page

**Actor:** Admin users

**Output:** Patient detail registration, viewing and management (patient discharge and transfers) page

The screenshot shows a web application interface for adding patient details. On the left is a navigation sidebar with a 'NAVIGATION' header and several menu items: 'Dashboard', 'Patients' (which is expanded to show 'Register Patient', 'View Patients', and 'Manage Patients'), 'Discharge Patients', 'Patient Transfers', 'Employees', 'Pharmacy', and 'Accounting'. The main content area is titled 'Add Patient Details' and has a breadcrumb trail 'Dashboard > Patients > Add Patient'. Below the title, there is a form with the instruction 'Fill all fields'. The form contains several input fields: 'First Name' and 'Last Name' (both with placeholder text 'Patient's First Name' and 'Patient's Last Name' respectively), 'Date Of Birth' (with a date format placeholder 'DD/MM/YYYY') and 'Age' (with a placeholder 'Patient's Age'), and a single-line 'Address' field (with a placeholder 'Patient's Address'). At the bottom of the form, there are three fields: 'Mobile Number', 'Patient Ailment', and 'Patient's Type' (which is a dropdown menu with 'Choose' as the selected option). A blue 'Add Patient' button is located at the bottom left of the form area.

**Fig. 4.5:** Patients details page

Flow:

1. Admin users can browse this page.
2. Admin users can view all modules here



## 4.5.5 Employee details page

**Actor:** Admin users

**Output:** Employee details (add, view, manage, assign and transfer) page

The screenshot displays the 'Add Employee Details' page. On the left is a navigation sidebar with a 'NAVIGATION' header. It contains links for 'Dashboard', 'Patients', 'Employees' (which is expanded to show 'Add Employee', 'View Employees', and 'Manage Employees'), 'Assign Department', 'Transfer Employee', 'Pharmacy', and 'Accounting'. The main content area is titled 'Add Employee Details' and includes a breadcrumb trail 'Dashboard > Employee > Add Employee'. The form contains the following fields: 'First Name' and 'Last Name' (each with a text input), 'Email' (with a text input), and 'Password' (with a text input). A green 'Add Employee' button is positioned below the password field. A mouse cursor is visible over the 'Email' input field.

**Fig. 4.6:** Employee details page

Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

## 4.5.6 Pharmacy details page

**Actor:** Admin users

**Output:** Pharmacy details (Add, view and manage pharm category, add, view and manage pharmaceuticals, add, view and manage prescriptions) page.

The screenshot displays a web application interface for managing pharmaceuticals. On the left is a sidebar menu with categories: Employees, Pharmacy (selected), and Accounting. Under Pharmacy, there are sub-items: Add Pharm Category, View Pharm Category, Manage Pharm Category, Add Pharmaceuticals, View Pharmaceuticals, and Manage Pharmaceuticals. Under Accounting, there are: Add Prescriptions, View Prescriptions, and Manage Prescriptions. The main content area is titled 'Update #465931288 - Test Pharma' and includes a breadcrumb trail: Dashboard > Pharmaceuticals > Manage Pharmaceutical. The form contains several input fields: 'Pharmaceutical Name' with the value 'Test Pharma', 'Pharmaceutical Quantity(Cartons)' with the value '36', 'Pharmaceutical Description' with the text 'This is a demo test. This is a demo test. This is a demo test.', 'Pharmaceutical Vendor', and 'Pharmaceutical Category'. A 'Fill all fields' label is positioned above the first two fields. The description field has a rich text editor toolbar with icons for bold, italic, bulleted list, numbered list, link, unlink, and help.

**Fig. 4.7:** Pharmacy details page

Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

## 4.5.7 Accounting details page

**Actor:** Admin users

**Output:** Accounting management page (Add and manage accounts payable, Add and manage accounts receivable)

The screenshot displays the 'Payable Account Details' page. On the left is a navigation sidebar with categories: Dashboard, Patients, Employees, Pharmacy, Accounting (selected), and Inventory. Under 'Accounting', there are links for 'Add Acc. Payable', 'Manage Acc. Payable', 'Add Acc. Receivable', and 'Manage Acc. Receivable'. The main content area is titled 'Payable Account Details' and includes a breadcrumb trail: 'Dashboard > Accounting > Manage Payable Account'. Below the title, there's a section 'Fill all fields' with two input fields: 'Account Name' (containing 'Test Account Name') and 'Account Amount(\$)' (containing '1100'). Below these is an 'Account Description' section with a rich text editor containing the text 'This is a demo test'. At the bottom of the form is an orange 'Update Account' button.

**Fig. 4.8:** Accounting details page

Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

## 4.5.8 Inventory page

**Actor:** Admin users

**Output:** Inventory page (inventory details of pharmaceuticals and assets)

Dashboard > Inventory > Pharmaceuticals Inventory

Pharmaceuticals Inventory

Search

#	Pharmaceutical Name	Pharmaceutical Barcode	Pharmaceutical Vendor	Pharmaceutical Category	Pharmaceutical Quantity	Action
1	Paracetamol	134057629	Dawa Limited Kenya	Antipyretics	500 Cartons	<a href="#">View</a>
2	Aspirin	452760813	Cosmos Kenya Limited	Analgesics	500 Cartons	<a href="#">View</a>
3	Test Pharma	465931288	Cosmos Pharmaceutical Limited	Antibiotics	36 Cartons	<a href="#">View</a>

1

**Fig. 4.9:** Inventory page

Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

## 4.5.9 Reporting page

**Actor:** Admin users

**Output:** Reporting page (Record maintenance of Inpatients, Outpatients, employees and pharmaceuticals)

NAVIGATION

- Dashboard
- Patients
- Employees
- Pharmacy
- Accounting
- Inventory
- Reporting
- InPatient Records
- OutPatient Records
- Employee Records
- Pharmaceutical Records

View Medical Records

Dashboard > Reporting > Medical Records

Search

#	Patient Name	Patient Number	Address	Ailment	Age	Action
1	Lawrence Bischof	ISL1E	82 Bryan Street	Demo Test	32 Years	<a href="#">View</a>
2	Michael White	DCRIB	60 Radford Street	Demo Test	30 Years	<a href="#">View</a>
3	Cynthia Connolly	3Z14K	9 Hill Haven Drive	Demo Test	22 Years	<a href="#">View</a>
4	John Doe	RAV6C	12 900 Los Angeles	Malaria	35 Years	<a href="#">View</a>

« 1 »

**Fig.5:** Reporting page

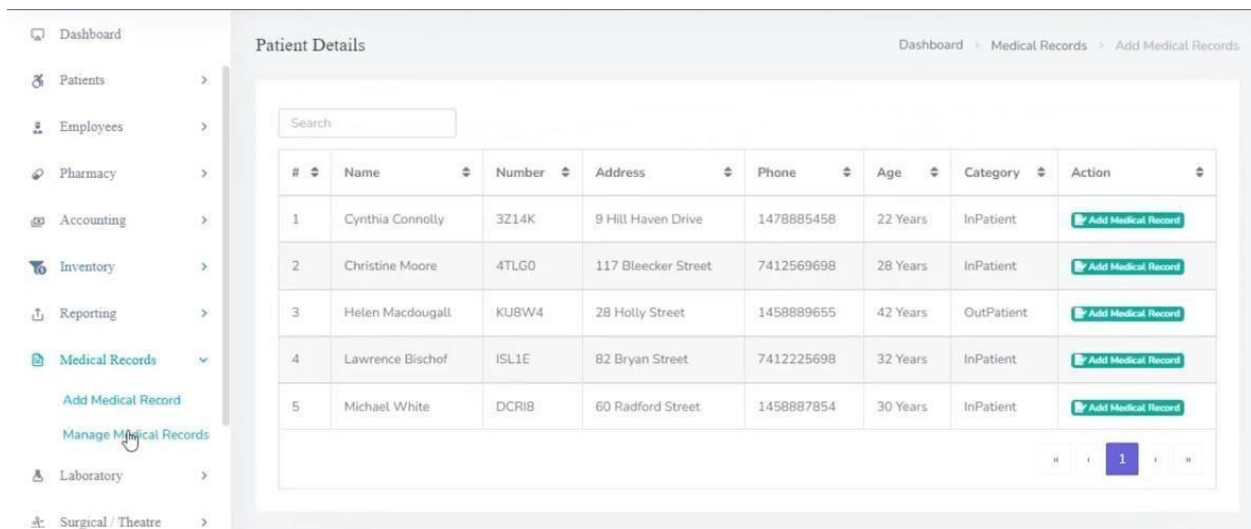
Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

### 4.5.1.1 Medical records page

**Actor:** Admin users

**Output:** Medical record maintenance page (Add and manage medical records)



**Fig. 5.1:** Medical records page

Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

### 4.5.1.2 Laboratory details page

**Actor:** Admin users

**Output:** Laboratory details maintenance page (Lab tests, lab reports, patient and employee vitals, lab reports, adding and managing lab equipment's)

**Laboratory** ▾

- Patient Lab Tests
- Patient Lab Results
- Patient Vitals
- Employee Vitals
- Lab Reports
- Add Lab Equipment
- Manage Lab Equipments

Surgical / Theatre >

Payrolls >

Vendors >

Password Resets >

Capture Christine Moore Vitals

Dashboard > Laboratory > Capture Vitals

Fill all fields

Patient Name: Christine Moore

Patient Ailment: Demo Test

Patient Number: 4TLG0

Patient Body Temperature °C: °C

Patient Heart Pulse/Beat BPM: HeartBeats Per Minute

Patient Respiratory Rate bpm: Breathes Per Minute

Patient Blood Pressure mmHg: mmHg

Add Vitals

**Fig. 5.2:** Laboratory details page

Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

### 4.5.1.3 Surgical / Theatre page

**Actor:** Admin users

**Output:** Surgical equipment's management and patient surgery details management

Pharmacy >  
Accounting >  
Inventory >  
Reporting >  
Medical Records >  
Laboratory >  
**Surgical / Theatre** >  
Add Equipment  
Manage Equipments  
**Add Patient**  
Manage Patients  
Surgery Records

Dashboard > Surgery | Theatre > Add Patient

Search

#	Patient Name	Patient Number	Patient Address	Patient Category	Action
1	Cynthia Connolly	3Z14K	9 Hill Haven Drive	InPatient	<a href="#">View</a> <a href="#">Add Patient</a>
2	Michael White	DCRi8	60 Radford Street	InPatient	<a href="#">View</a> <a href="#">Add Patient</a>
3	Lawrence Bischof	ISL1E	82 Bryan Street	InPatient	<a href="#">View</a> <a href="#">Add Patient</a>
4	Christine Moore	4TLG0	117 Bleecker Street	InPatient	<a href="#">View</a> <a href="#">Add Patient</a>
5	Helen Macdougall	KU8W4	28 Holly Street	OutPatient	<a href="#">View</a> <a href="#">Add Patient</a>

1

**Fig. 5.3:** Surgical / Theatre page

Flow:

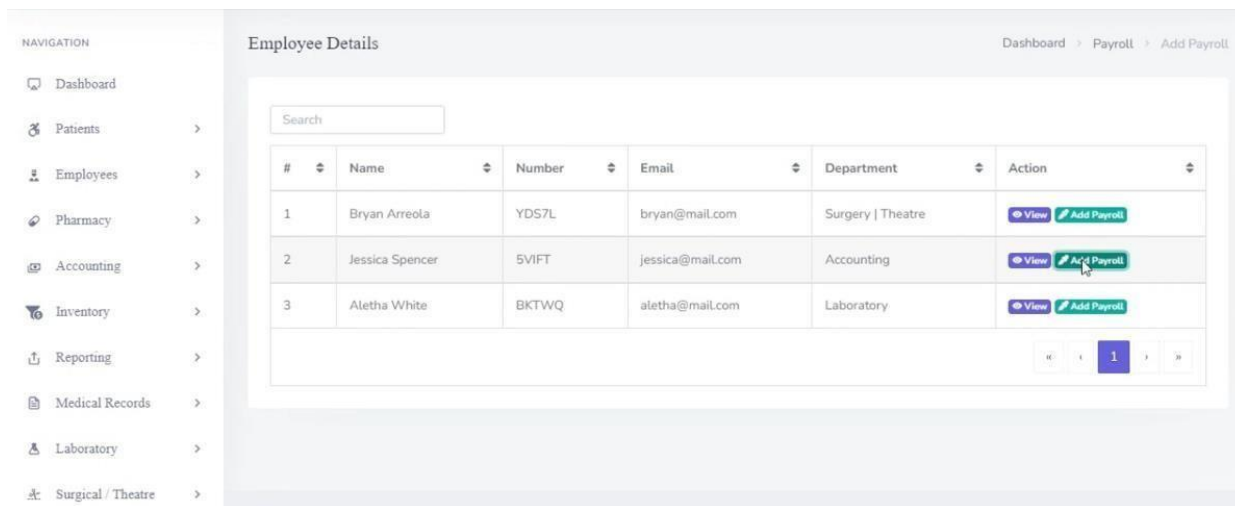
1. Admins can browse this page.
2. Admin users can view all the modules here.



#### 4.5.1.4 Payroll management page

**Actor:** Admin users

**Output:** Payroll management page (Add and manage payrolls)



**Fig. 5.4:** Payroll management page

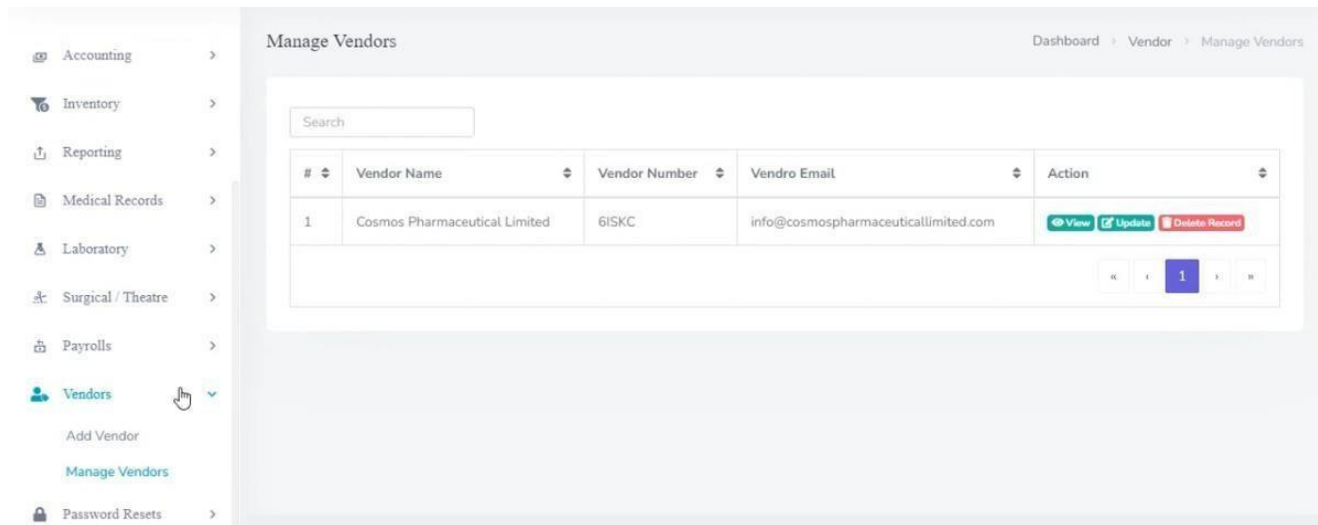
Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

#### 4.5.1.5 Vendor management page

**Actor:** Admin users

**Output:** Vendor management page (Add and manage vendors)



**Fig. 5.5:** Vendor management page

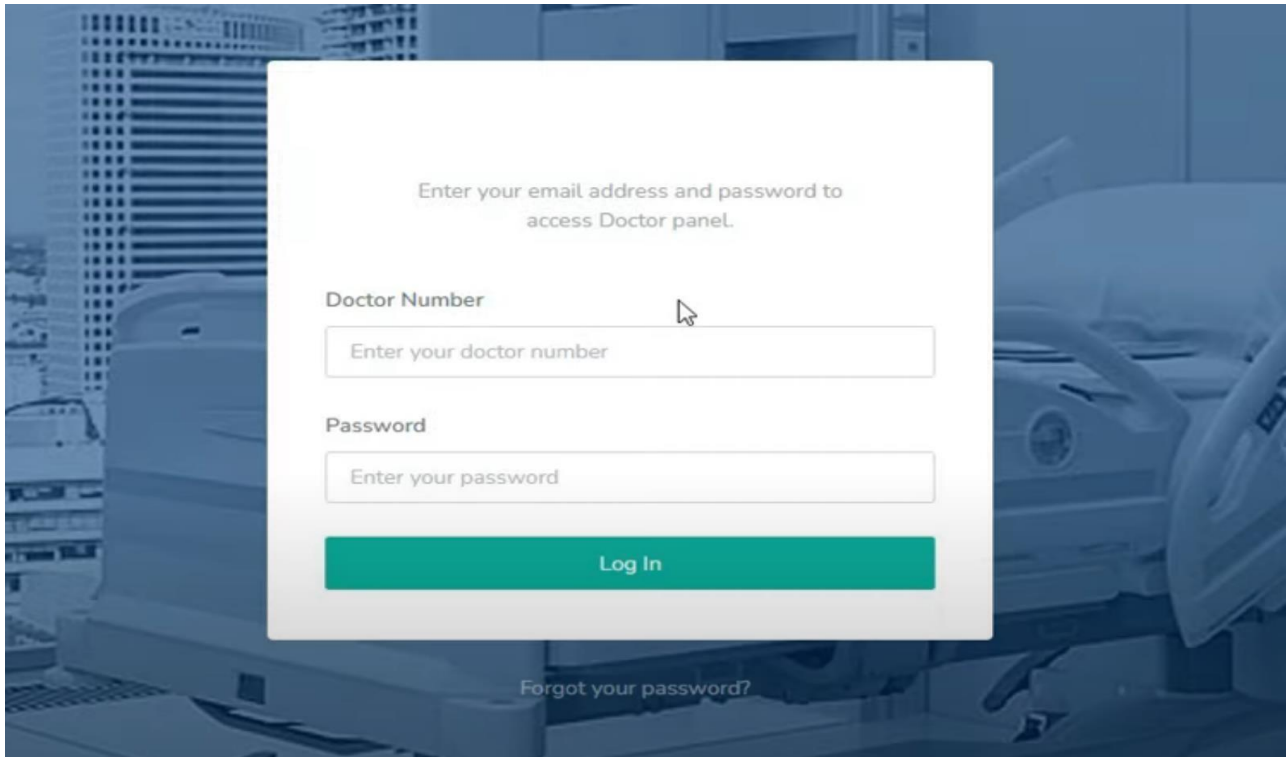
Flow:

1. Admins can browse this page.
2. Admin users can view all the modules here.

#### 4.5.1.6 Doctor's login page

**Actor:** The doctor or specialist

**Output:** Login page for doctors



**Fig. 5.6:** Doctor's login page

Flow:

1. Only doctor's user can browsers this page.
2. Doctor's can search all patient appointment and all users' activities.

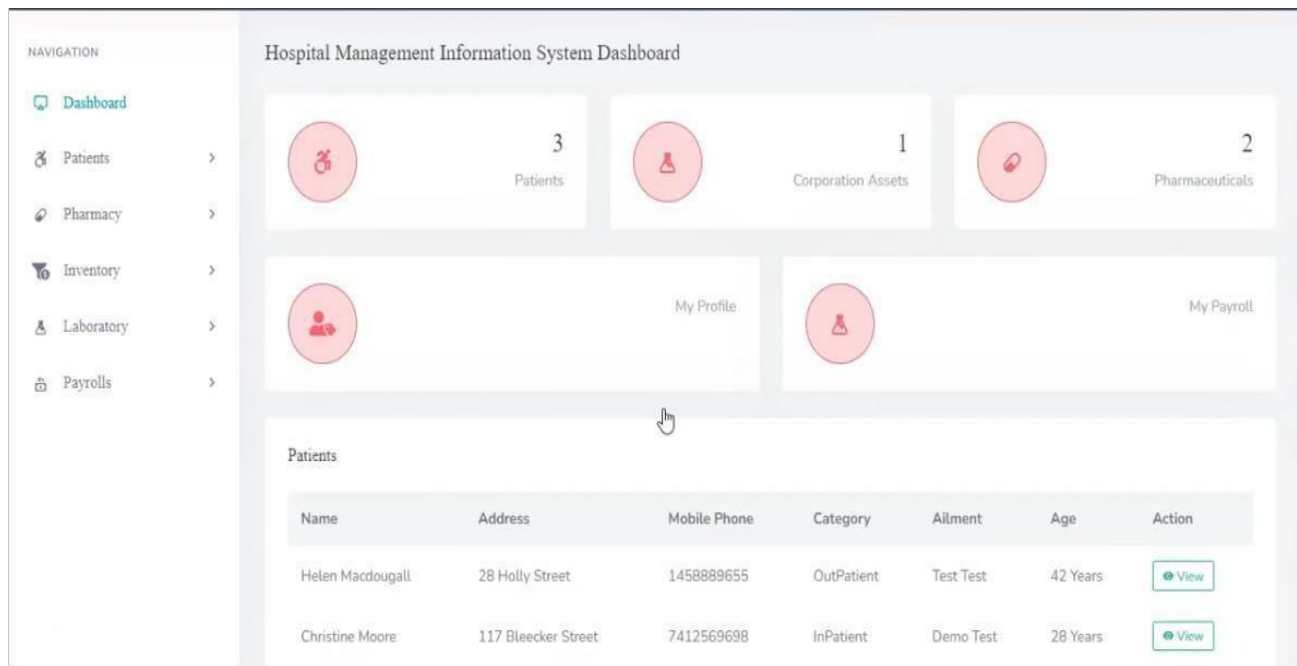
Alternate Flow:

- (3) If the user email is wrong then it is asked to login again.
- (4) If the password is wrong then the user is asked to enter again.

#### 4.5.1.7 Doctor's dashboard page

**Actor:** The doctor or specialist

**Output:** Doctor's dashboard page



**Fig. 5.7:** Doctor's dashboard page

Flow:

1. Only doctor's user can browsers this page.
2. Doctor's can search all patient appointment and all users' activities.

## **Chapter Five**

### **5. System Implementation**

#### **5.1 Implementation**

Implementation is the process of having system personal check out and provides new equipment's into use, train the user to install a new application and construct any files of data needed to use it. There are three types of implementations. Implementation of computer system to replace a manual system. To problem encountered are covering files, training user, creating accurate files and verifying print outs for integrity. Implementation of a new computer system to replace an existing one. This is usually difficult conversion. If not properly planned, there can be many problems. So large computer system many take as long as a year to convert. Implementation of a modified application to replace the existing one using the same computer. This type of conversing is relatively easy to handle, usually there are no major change in the file. Our project is yet to be implemented.

#### **5.2 Implementation Environment**

The implementation view of software requirement presents the real-world manifestation of processing functions and information structures. This computerized system is specified in a manner that dictates accommodation of certain implementation details.

The implementation environment of the developed system facilitates multiple users to use this system simultaneously. The user interfaces are designed keeping in mind that the users of this system are familiar to using GUI-based systems. Thus, we restricted ourselves to developing a GUI-based system so that it becomes easier for the end user to get acquainted to the developed system.

## **5.3 Functional Requirement**

This system interface is divided into two sections

1. Administrator interface.
2. Users interface.

### **5.3.1 Administrator Interface**

1. Administrator can delete any post.
2. Administrator can verify user account.

### **5.3.2 User Interface**

1. User can browse all ads without any account.
2. For post an ad needs to create an account
3. User can update/edit their own account.
4. Log in and log out system.
5. To create a new account user must be needs to verify his email with verification code.
6. If any user forgets his/her password he/she can recovery his account with verify his email and create a new password.

### **5.3.3 Login Information:**

Log in details

Database Name: hmisphp

Developed by srinivasan.m

Recommended PHP Version 5.6, 7.4

Admin Login Details

Email : admin@mail.com

Password: Password@123

Doctor Login Details

ID : YDS7L

Password: password

### **5.3.4 Available features:**

The available features are:

- Admin Panel
- Employee Panel
- Patient Management
- Transfer, Discharge Patient
- Employee Management
- Assign Departments
- Transfer Employees
- Medicine Management
- Prescriptions
- Accounting – Payable and Receivable
- Inventory Management
- Assets Management
- In and Out Patient Records
- Medical Records
- Patient Lab Test and Results
- Manage Patient's Vitals

- Lab Reports
- Surgery Records
- Surgery Equipment Records
- Patient Medical Profile
- Payroll Management
- Print Payroll Receipt
- Manage Vendors

### **5.3.5 Programs used:**

Project Name: Hospital Management System

Language/s Used: PHP

PHP version (Recommended): 5.6, 7.4

Database: MySQL

Type: Web Application

Developer: Martin Mbithi Nzilani

Updates: 0



### 5.3.6 Source codes:

#### HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required Meta Tags -->
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <!-- Page Title -->
  <title>Hospital Management System</title>

  <!-- Favicon -->
  <link rel="shortcut icon" href="assets/images/logo/favicon.png" type="image/x-icon">

  <!-- CSS Files -->
  <link rel="stylesheet" href="assets/css/animate-3.7.0.css">
  <link rel="stylesheet" href="assets/css/font-awesome-4.7.0.min.css">
  <link rel="stylesheet" href="assets/css/bootstrap-4.1.3.min.css">
  <link rel="stylesheet" href="assets/css/owl-carousel.min.css">
  <link rel="stylesheet" href="assets/css/jquery.datetimepicker.min.css">
  <link rel="stylesheet" href="assets/css/linearicons.css">
  <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
  <!-- Preloader Starts -->
  <div class="preloader">
    <div class="spinner"></div>
  </div>
  <!-- Preloader End -->

  <!-- Header Area Starts -->
  <header class="header-area">
    <div id="header" id="home">
      <div class="container">
        <div class="row align-items-center justify-content-between d-flex">
          <div id="logo">
            <a href="index.php"></a>
```

```

</div>
<nav id="nav-menu-container">
  <ul class="nav-menu">
    <li class="menu-active"><a href="index.php">Home</a></li>
    <li><a href="backend/doc/index.php">Doctor's Login</a></li>
    <li><a href="backend/admin/index.php">Administrator Login</a></li>
  </ul>
</nav><!-- #nav-menu-container -->
</div>
</div>
</div>
</header>
<!-- Header Area End -->

<!-- Banner Area Starts -->
<section class="banner-area">
  <div class="container">
    <div class="row">
      <div class="col-lg-8">
        <h4>Caring for better life</h4>
        <h1>Medicines Cure diseases,but only doctors can cure Patients</h1>
        <p>srini is awarded as one of the Top Hospital Management System, which can integrate all the HIS
systems, processes and machines into an intelligent information system to derive operational efficiency and assist
hospitals in decision making process through MIS and Analytics.</p>
      </div>
    </div>
  </div>
</section>

```

### Javascript code :

```

<!-- Javascript -->
<script src="assets/js/vendor/jquery-2.2.4.min.js"></script>
  <script src="assets/js/vendor/bootstrap-4.1.3.min.js"></script>
<script src="assets/js/vendor/wow.min.js"></script>
<script src="assets/js/vendor/owl-carousel.min.js"></script>
<script src="assets/js/vendor/jquery.datetimepicker.full.min.js"></script>
<script src="assets/js/vendor/jquery.nice-select.min.js"></script>
<script src="assets/js/vendor/superfish.min.js"></script>
<script src="assets/js/main.js"></script>

</body>
</html>

```

## PHP Code:

```
<?php
    session_start();
    include('assets/inc/config.php');//get configuration file
    if(isset($_POST['doc_login']))
    {
        $doc_number = $_POST['doc_number'];
        //$doc_email = $_POST['doc_ea']
        $doc_pwd = sha1(md5($_POST['doc_pwd']));//double encrypt to increase security
        $stmt=$mysqli->prepare("SELECT doc_number, doc_pwd, doc_id FROM his_docs WHERE
doc_number=? AND doc_pwd=?");//sql to log in user
        $stmt->bind_param('ss', $doc_number, $doc_pwd);//bind fetched parameters
        $stmt->execute();//execute bind
        $stmt -> bind_result($doc_number, $doc_pwd, $doc_id);//bind result
        $rs=$stmt->fetch();
        $_SESSION['doc_id'] = $doc_id;
        $_SESSION['doc_number'] = $doc_number;//Assign session to doc_number id
        //$uip=$_SERVER['REMOTE_ADDR'];
        //$ldate=date('d/m/Y h:i:s', time());
        if($rs)
        {
            //if its sucessfull
            header("location:his_doc_dashboard.php");
        }

        else
        {
            #echo "<script>alert('Access Denied Please Check Your Credentials');</script>";
            $err = "Access Denied Please Check Your Credentials";
        }
    }
?>
<!--End Login-->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <title>SRINI Hospital Management System </title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta content="" name="description" />
    <meta content="" name="MartDevelopers" />
```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<!-- App favicon -->
<link rel="shortcut icon" href="assets/images/favicon.ico">

<!-- App css -->
<link href="assets/css/bootstrap.min.css" rel="stylesheet" type="text/css" />
<link href="assets/css/icons.min.css" rel="stylesheet" type="text/css" />
<link href="assets/css/app.min.css" rel="stylesheet" type="text/css" />
<!--Load Sweet Alert Javascript-->

<script src="assets/js/swal.js"></script>
<!--Inject SWAL-->
<?php if(isset($success)) { ?>
<!--This code for injecting an alert-->
    <script>
        setTimeout(function ()
        {
            swal("Success","<?php echo $success;?>","success");
        },
        100);
    </script>

<?php } ?>

<?php if(isset($err)) { ?>
<!--This code for injecting an alert-->
    <script>
        setTimeout(function ()
        {
            swal("Failed","<?php echo $err;?>","error");
        },
        100);
    </script>

<?php } ?>

</head>

<body class="authentication-bg authentication-bg-pattern">

<div class="account-pages mt-5 mb-5">
    <div class="container">

```

```

<div class="row justify-content-center">
  <div class="col-md-8 col-lg-6 col-xl-5">
    <div class="card bg-pattern">

      <div class="card-body p-4">

        <div class="text-center w-75 m-auto">
          <a href="index.php">
            <span></span>
          </a>
          <p class="text-muted mb-4 mt-3">Enter your email address and password to access
Doctor panel.</p>
        </div>

        <form method='post' >

          <div class="form-group mb-3">
            <label for="emailaddress">Doctor Number</label>
            <input class="form-control" name="doc_number" type="text" id="emailaddress"
required="" placeholder="Enter your doctor number">
          </div>

          <div class="form-group mb-3">
            <label for="password">Password</label>
            <input class="form-control" name="doc_pwd" type="password" required=""
id="password" placeholder="Enter your password">
          </div>

          <div class="form-group mb-0 text-center">
            <button class="btn btn-success btn-block" name="doc_login" type="submit"> Log
In </button>
          </div>

        </form>

        <!--
For Now Lets Disable This
This feature will be implemented on later versions
<div class="text-center">
  <h5 class="mt-3 text-muted">Sign in with</h5>
  <ul class="social-list list-inline mt-3 mb-0">
    <li class="list-inline-item">
      <a href="javascript: void(0);" class="social-list-item border-primary text-
primary"><i class="mdi mdi-facebook"></i></a>

```

```

        </li>
        <li class="list-inline-item">
            <a href="javascript: void(0);" class="social-list-item border-danger text-
danger"><i class="mdi mdi-google"></i></a>
        </li>
        <li class="list-inline-item">
            <a href="javascript: void(0);" class="social-list-item border-info text-info"><i
class="mdi mdi-twitter"></i></a>
        </li>
        <li class="list-inline-item">
            <a href="javascript: void(0);" class="social-list-item border-secondary text-
secondary"><i class="mdi mdi-github-circle"></i></a>
        </li>
    </ul>
</div>
-->

</div> <!-- end card-body -->
</div>
<!-- end card -->

<div class="row mt-3">
    <div class="col-12 text-center">
        <p> <a href="his_doc_reset_pwd.php" class="text-white-50 ml-1">Forgot your
password?</a></p>
        <!-- <p class="text-white-50">Don't have an account? <a href="his_admin_register.php"
class="text-white ml-1"><b>Sign Up</b></a></p>-->
    </div> <!-- end col -->
</div>
<!-- end row -->

</div> <!-- end col -->
</div>
<!-- end row -->
</div>
<!-- end container -->
</div>
<!-- end page -->

<?php include ("assets/inc/footer1.php");?>

<!-- Vendor js -->
<script src="assets/js/vendor.min.js"></script>

```

```
<!-- App js -->  
<script src="assets/js/app.min.js"></script>  
  
</body>  
  
</html>
```

## **Chapter Six**

### **6. System Testing**

#### **6.1 Integration Testing**

Integration testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each modules effect on the entire program model. After integration testing the project works successfully.

#### **6.2 Unit Testing**

Unit testing performed on each module or block of code during development. Unit testing is normally done by the programmer who writes the code.

#### **6.3 System Testing**

System testing done by a professional testing agent on the completed software product before it is introduced to the market.

#### **6.4 Acceptance Testing**

Acceptance testing is a beta testing of the product done by the actual end user.

#### **6.5 Recovery Testing**

Recovery testing is done to demonstrate a software salutation is reliable, trustworthy and can successfully recoup form possible crashes.



## **6.6 Functional Testing**

Functional Testing also known as functional completeness testing. Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could to improve it during functional testing.

## **6.7 Hardware/Software Testing**

IBM refers to Hardware/Software testing as “HW/SW Testing”. This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing.

## **6.8 Security Testing**

Security Testing is a variant of Software Testing which ensures, that system and applications in an organization, are free from any loopholes that may cause a big loss. Security testing of any system is about finding all possible loopholes and weaknesses of the system which might result into a loss of information at the hands of the employees or outsiders of the Organization

## **6.9 Advantages**

The software helps to handle the entire administration of hospitals and healthcare facilities. Typically, such a software includes various modules that help doctors manage their assignments and schedules, carry out patient registration, maintain store inventory records, keep track of medicine, administration, maintain blood bank (with available blood type) details, individual record of patients with their test reports, nursing and housekeeping service details, financial information, including final billing & payments, insurance details and much more. After the customized software is implemented and integrated into the system, patient care and hospital administration becomes an easy job.

## **Chapter seven**

### **7. Conclusion**

#### **7.1 Conclusions**

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

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

#### **7.2 Limitations of the system**

- Online payment is not available at this version.
- Data delete & edit system is not available for all section.
- User account not verified by Mobile SMS not available in this system.
- Loss of data due to mismanagement.

## References

- [1] Deepak Thomas "Beginning PHP 4 Databases", Wrox Press Ltd. Paperback- 17, October, 2002. 70-130 pp.
- [2] Matt Doyle, "Beginning PHP 5.3, 2<sup>nd</sup> edition", October 2009. 150-270 pp.
- [3] Luke Welling, Laura Thomson. Sams HP and MySQL Web Development, 2<sup>nd</sup> edition, Paperback- 20 February, 2003. 105-209 pp.
- [4] W. Jason Gilmore "Beginning PHP 5 and MySQL 5 from Novice to Professional SECONDEDITION", Jul 9, 2008. 100-150 pp.
- [5] Abraham Silberschatz, Henry F. Korth and S. Sudarshan "Sixth Edition Database System Concepts released", January 28, 2010. 206-253 pp.
- [6] Server-Side Scripting <http://php.net/manual/en/index.php>, Last accessed on 12/15/2017 at 2:33pm
- [7] HTML & CSS <https://www.w3schools.com/>, Last accessed on 10/21/2017 at 1:33pm.
- [8] Bootstrap <http://getbootstrap.com/>, last accessed on 09/30/2017 at 10:10pm.
- [9] <https://stackoverflow.com/>, last accessed on 11/07/2017 at 2:20am.