



# System analysis and design, SCS 2104 project

Lecturer : *Mrs S. Dube*

Students :

- *Lorthar Mabandla , N0176543M*
- *Qhelile Sibanda , N0171603M*
- *Tendai D. Zhou , N0175499D*
- *Ashton Tande , N0172470F*

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>INTRODUCTION</b>	<b>4</b>
2.1	Introduction . . . . .	4
2.2	Problem Identification . . . . .	4
2.2.1	Further Problem Classification . . . . .	6
2.3	Modules In The Project . . . . .	6
2.3.1	Admin/Receptionist Module . . . . .	6
2.3.2	Client/User Module . . . . .	7
2.3.3	Doctor Module . . . . .	7
<b>3</b>	<b>REQUIREMENTS SPECIFICATION</b>	<b>8</b>
3.1	INTRODUCTION . . . . .	8
3.2	HARDWARE REQUIREMENTS . . . . .	8
3.2.1	HARDWARE REQUIREMENTS FOR THE PRESENT PROJECT . . . . .	8
3.3	SOFTWARE REQUIREMENTS . . . . .	8
3.3.1	SOFTWARE REQUIREMENTS FOR PRESENT PROJECT . . . . .	9
<b>4</b>	<b>ANALYSIS</b>	<b>9</b>
4.1	Existing System . . . . .	9
4.2	Organogram . . . . .	10
4.3	Data Flow Diagram For The Old System . . . . .	11
4.4	Surgery Management(Old) Functional-Decompositional-Diagram . . . . .	12
4.5	Possible Solutions . . . . .	12
4.6	Proposed System . . . . .	13
4.6.1	Proposed Solution Justification . . . . .	13
4.7	Future Work . . . . .	13
4.8	Feasibility Study . . . . .	14
4.8.1	Economic Feasibility . . . . .	14
4.8.2	Technical Feasibility . . . . .	14
4.8.3	Operational Feasibility . . . . .	14
4.9	Software Specification . . . . .	14
<b>5</b>	<b>Project Methodology</b>	<b>20</b>
5.1	Reasons For Choosing The Iterative Method . . . . .	20
<b>6</b>	<b>DESIGN</b>	<b>25</b>
6.1	System Design . . . . .	25
6.1.1	Introduction to Functional Decomposition Diagram(FDDs) . . . . .	25
6.1.2	FDDs Of The Project . . . . .	25
6.1.3	Introduction to Data Flow Diagrams(DFDs) . . . . .	26
6.1.4	FDDs Of The Project . . . . .	28

6.1.5	Introduction to Entity-Relationship(ER) Diagrams . . . . .	34
6.1.6	Entity-Relationship Diagram for The Project . . . . .	36
<b>7</b>	<b>SYSTEM IMPLLEMENTATION</b>	<b>37</b>
7.1	Introduction . . . . .	37
7.2	Sample code . . . . .	37
7.2.1	Add Doctor PHP . . . . .	37
7.2.2	Admin Panel PHP . . . . .	38
7.2.3	Appointment PHP . . . . .	39
7.2.4	Search PHP . . . . .	40
7.2.5	Update PHP . . . . .	43
7.2.6	Logout PHP . . . . .	44
7.2.7	Error PHP . . . . .	45
7.2.8	Surgery Management Database System SQL . . . . .	47
7.2.9	Func PHP . . . . .	49
7.2.10	STYLE CSS . . . . .	55
7.2.11	Index PHP . . . . .	56
<b>8</b>	<b>TESTING</b>	<b>59</b>
8.1	Introduction . . . . .	59
8.2	Testing Methods . . . . .	59
<b>9</b>	<b>SAMPLE SCREENSHOTS</b>	<b>62</b>
9.1	Login . . . . .	62
9.2	Appointment . . . . .	63
9.3	New Doctor . . . . .	63
9.4	Payment Status . . . . .	64
9.5	Result Searching One . . . . .	64
9.6	Result Searching Two . . . . .	65
<b>10</b>	<b>CONCLUSION</b>	<b>66</b>

# 1 Abstract

Medical surgeries are eventful organizations with many activities taking place concurrently. As a result of this they need fairly robust dedicated systems to keep track of information and activities for example patient records, medical records, licenses and research documents. This information might not only be merely stored, but might also need to be manipulated in order to make it useful. One of the main concerns of this system is end user privacy (patient privacy). Various measures have been taken in this project in order to mitigate the risk of patient information falling into the wrong hands, namely:

1. **A timeout clock:**

This clock logs the user out of the system when a certain amount of time has passed and they have not interacted with the system in any way. This helps keep the patient information private in case they forgot to log out.

2. **End to end encryption**

We made use of the Diffie Hellman encryption algorithm in order to combat against cyber threats such as hackers.

This patient record management system makes record keeping and auditing of the surgery less tedious, by offering automated services such as online patient registration and computerised billing system. Since the information is stored in a database this makes modern concepts such as data mining possible and this has the potential to help in the medical research field.

## 2 INTRODUCTION

### 2.1 Introduction

The project Surgery Management system includes registration of patients, storing their details into the system, and also computerized billing. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

The Surgery Management System can be entered using a username and password. It is accessible by all the users that is the administrator or receptionist, the doctor and the clients. The administrator or receptionist and the doctor can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

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

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

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

### 2.2 Problem Identification

1. **Lack of information retrievals:-** The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various receipts to check how long has the client been attending to the *Surgery*. This results in inconvenienced wastage of time.
2. **Lack of immediate information storage:-** The information generated by various transactions takes time and efforts to be stored at right place.
3. **Lack of prompt updating:-** Various changes to information like patient medication details or are difficult to make as paper work is involved (regularly bought prescriptions).

4. **Error prone manual calculation:-** Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient's bill based on various treatments.
5. **Preparation of accurate and prompt reports:-** This becomes a difficult task as information is difficult to collect from various registers.
6. **Objective:-**
  - Define Surgery
  - Recording information about the Patients that come.
  - Generating bills.
  - Recording information related to diagnosis given to Patients.
  - Keeping records of the Immunization provided to children/patients.
  - Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in the Surgery by the operational staff and Doctors. Most of these works haven't been done and if done they are done on paper.

#### 7. **Scope of the Project:-**

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

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can not remember them at that time.

### **2.2.1 Further Problem Classification**

As we have observed the way the Surgery functions we discover that their firm is still functioning well but the doctors have a problem during their daily operations as they have no proper way of recording their clients and making follow up on the clients. They have do not have adequate information about their clients for example their place of residence. The firm just gives receipts to the clients of how much the particular procedure or items cost. So if the firm decides to open a new branch, they will not know the residential locations of their clients which would be useful in selecting a central site for the new branch. Information such as demographics help the Surgery serve its customer to the fullest potential and thereby achieving a high customer satisfaction rate and healthy profit in terms of business.

The other thing we observed during their daily operation is that they have no way of grouping clients according to age as they don't have the essential clients information. As this information has been seen to be useful over the years to determine which age group will be affected given that there is a disease outbreak. Information regarding age groups is also useful in analysing trends of the infections, with this information, the firm will easily identify its target market within a time frame of the disease outbreaks. Also this information is crucial for their advertisement depending on products regarding particular age groups.

The firm has no proper system to book appointments with the doctors in advance so there is a bottleneck. This has led to poor services as most of their clients will have to stay long periods before they get the doctors attention. With this kind of bottleneck it tends to reduce the firm's services meaning that their services will be rendered slow. The reason that most people who prefer going to a surgery is because there is less bottleneck, but if also the surgery has bottlenecks it means their clients might return back to hospitals as there will be a mere difference in the way of conducting their services compared with hospitals.

## **2.3 Modules In The Project**

The entire project mainly consists of 3 modules, which are

1. **Admin/Receptionist Module**
2. **Client/User Module**
3. **Doctor Module**

### **2.3.1 Admin/Receptionist Module**

- manage department of surgery, client/user, doctor accounts.
- watch appointment of doctors
- watch transaction reports of patient payment

- Bed ,ward, cabin status
- watch blood bank report
- watch medicine status of surgery stock
- watch operation report
- watch birth report
- watch diagnosis report
- watch death report

### **2.3.2 Client/User Module**

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

### **2.3.3 Doctor Module**

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



## **3 REQUIREMENTS SPECIFICATION**

### **3.1 INTRODUCTION**

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

### **3.2 HARDWARE REQUIREMENTS**

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

#### **3.2.1 HARDWARE REQUIREMENTS FOR THE PRESENT PROJECT**

Processor : Dual core processor, at least 1.6Ghz

Main memory : 2 Giga-bytes

Hard disk : Greater than 25 Giga-bytes

### **3.3 SOFTWARE REQUIREMENTS**

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

### 3.3.1 SOFTWARE REQUIREMENTS FOR PRESENT PROJECT

Operating system : WINDOWS 7,8,10

Software : XAMP/WAMP , Simple text editor such as Atom

DBMS : MYSQL relational database management system

## 4 ANALYSIS

### 4.1 Existing System

The chosen organization is a surgery that is being operated by two doctors, husband and wife and a few other general workers. The receptionist lets the clients know when the doctor is free. The receptionist also has the role of charging people and accepting payments. The doctors only treat the clients and after checking the proof of payments that have been waived by the receptionist. The doctors perform the management of the surgery which includes necessary paperwork after normal working hours.

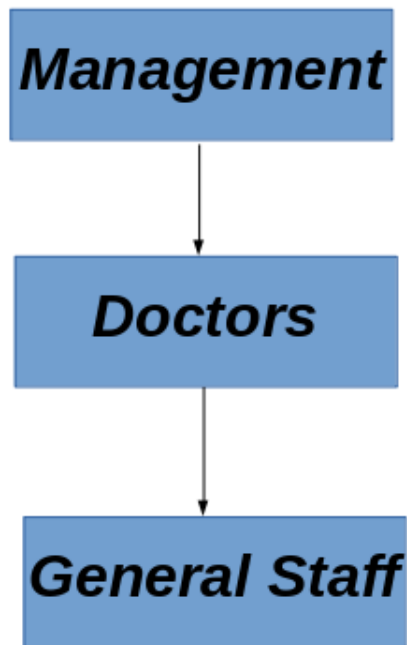
The receptionist just records the client's name and surname and the service they are purchasing, this information is recorded on a receipt for the purposes of auditing by the management and also as proof of payment when the client gets a session with the doctor. The doctor can be booked on a different book for the purposes of a surgery upon recommendation by the doctor. As a result there tends to be a bottleneck and clients tend to be in a long queue while waiting to be served by the doctor. There are slim chances that a doctor can make a follow up on a particular client to keep tabs on the recovery process of the client unless the client goes to see the doctor. Client contact details are barely recorded except in the event that a client is booking a slot for a surgery operation. Clients that come for diagnosis of minor sicknesses such as running stomachs and other related cases, their contact details and diagnosis results are barely kept for future reference and use by the doctors. However, it is these minor sicknesses that amount to a critical illness at some point, hence the need for a readily available patient medical history before diagnosis.

The technique used to understand the current method of implementation of the system was through observations as potential clients. We also interviewed a few clients from the surgery. We decided to use this approach as we realized workers had very limited time to entertain our questions as they were working on the clock and several patients needed to be attended to at a given time.

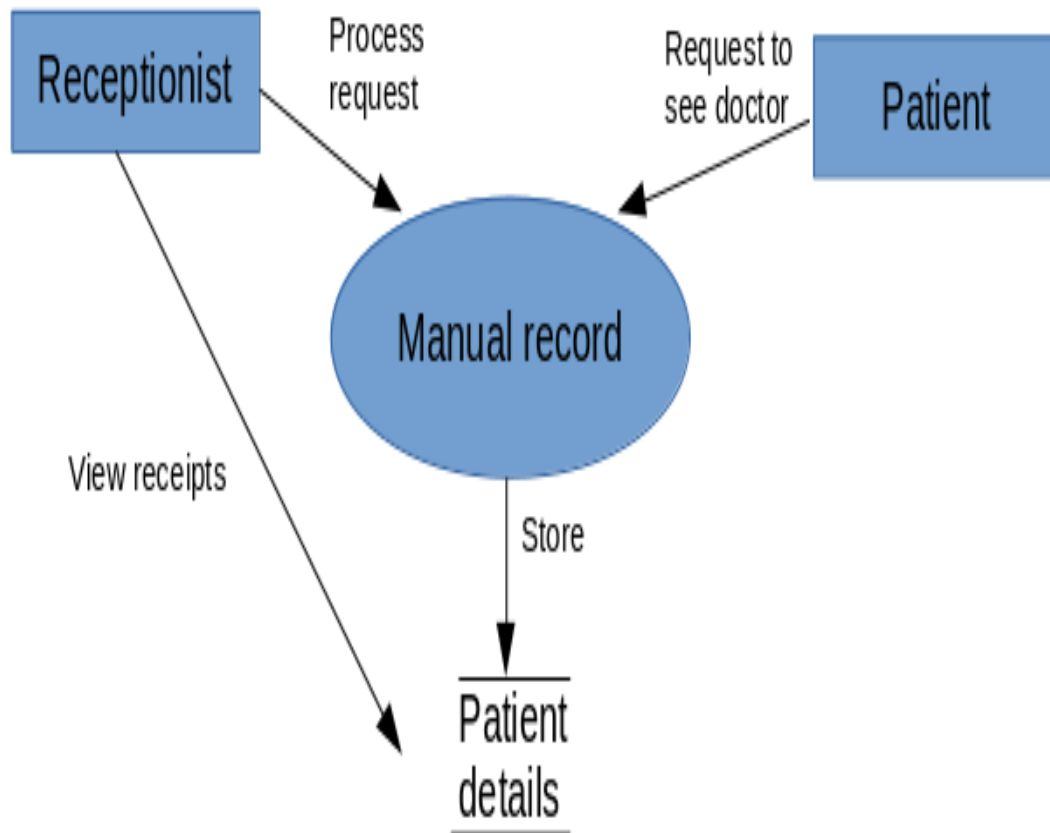
The observation technique was crucial during our problem finding phase as this is enabled us to view the operation of the surgery from the patients view although the view was based on our thoughts. However, the observation technique tends to be biased as only one side of the view is portrayed.

We also used interviews during our problem finding phase. This turned out to be very useful as it gave us views from both the surgery's view (in form of workers) and the view of the patients. This also enabled us to visualize the thoughts and concerns of the people we interviewed. However, at some points we felt that the information we were given by the surgery's staff was highly filtered as they would not want to taint the company's image. Moreover, admitting your weaknesses is not an easy thing to do hence we could understand their point of view.

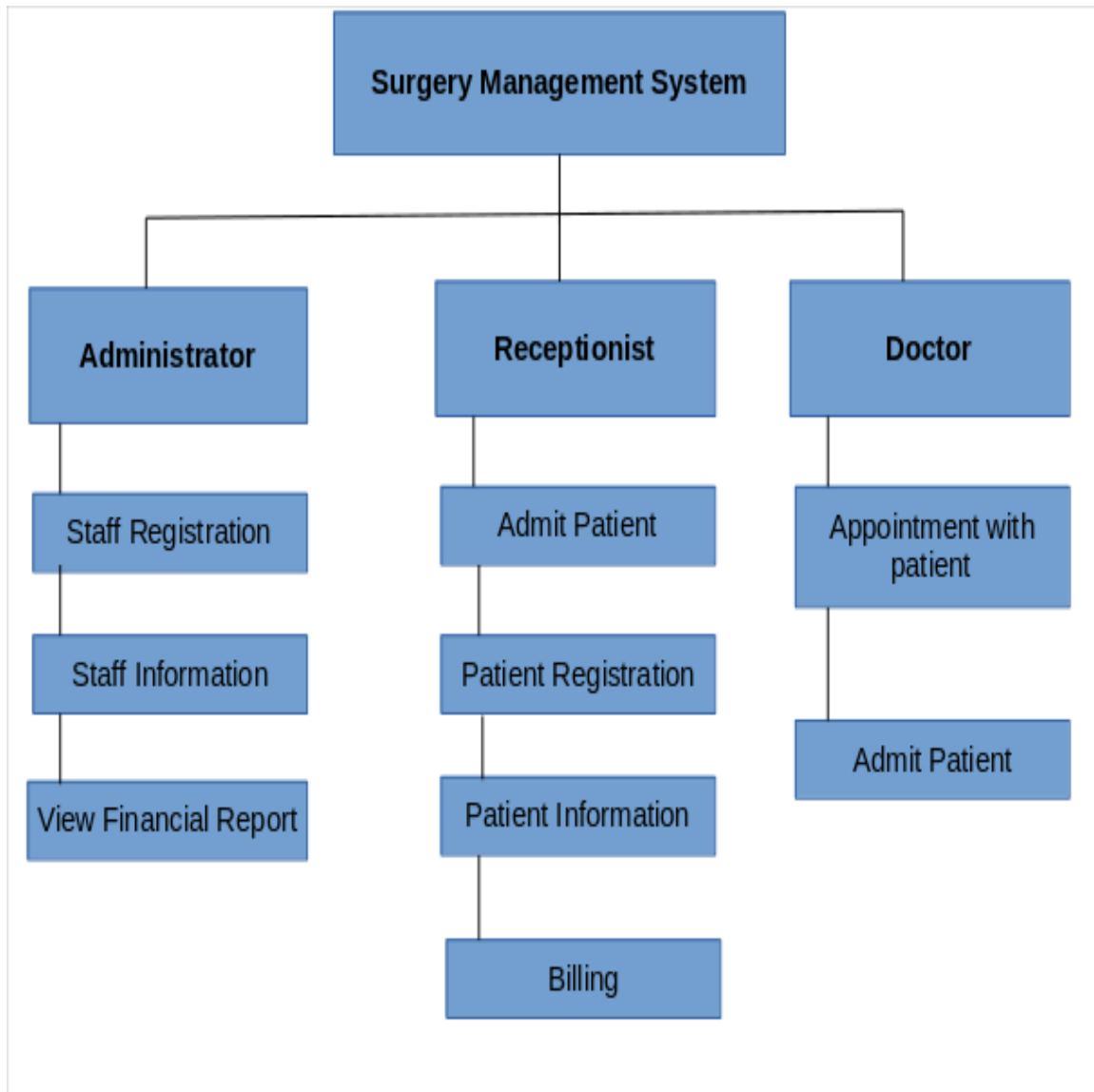
## 4.2 Organogram



### 4.3 Data Flow Diagram For The Old System



#### 4.4 Surgery Management(Old) Functional-Decompositional-Diagram



#### 4.5 Possible Solutions

1. Java Stand Alone Application(Unix and Dos Based Systems)
2. Smart Phone Mobile Application(Android and IOS)
3. Smart Watch Application
4. Web Based System

## 4.6 Proposed System

Due to the discovery that has been made during the observation stage, we decided to create a system that will help the firm to be able to reduce the bottleneck within the firm and should be able capture clients information and be able to classify the clients into related clinical groups. Thus the system should capture the following details:

*Name, Surname, ID number, Gender, Date of birth, Treatment, Booking information, Home address, Payment information.*

We will also create an online booking system to avoid bottlenecks of booking with the operational environment of the firm.

### 4.6.1 Proposed Solution Justification

#### Reasons For Choosing the System

- We choose the Web Based system because it interacts with the Database in real time. When a change is made to the system, all relevant users can readily notice the change soon after refreshing the web page.
- Facilitates remote access to the system for all users for example the doctor, patient and the support team.
- The developers can access and modify the system without going to the physical location of the system.
- There are readily available security features such as ssl, https.
- Since the system is online there is no need for sophisticated machinery such as servers, routes, access points and therefore costs are reduced.

## 4.7 Future Work

- **Implementing Patient User Interface**
- **Group Patients:** According to their place of residents. This is useful when determining where to open a new branch based on the number of clients in that particular area.

According to their prognosis. This is useful in determining disease trends or even for medical research.

- **Develop a Public Website:** That enables the general public to view their information updates, services and contact details.
- **Android Application**
- **Smart Watch Application**

## 4.8 Feasibility Study

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

Three key considerations involved in the feasibility analysis are:

### 4.8.1 Economic Feasibility

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

### 4.8.2 Technical Feasibility

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

### 4.8.3 Operational Feasibility

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

## 4.9 Software Specification

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

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like `<html>`). HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent empty elements and so are unpaired, for example `<img>`. The first tag in a pair is the start tag, and the second tag is the end tag (they

are also called opening tags and closing tags). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

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

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

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

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

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

- **MySQL:**

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

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



ically.

- **FEATURES OF MySQL**

**Internals and portability:**

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

**Security**

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

**Scale-ability and Limits:**

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

- **CONNECTIVITY:**

Clients can connect to MySQL Server using several protocols:

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

- **LOCALIZATION:**

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

- **CLIENTS AND TOOLS**

- MySQL includes several client and utility programs. These include both command-line programs such as mysqldump and mysqladmin, and graphical programs such as MySQL Workbench.
- MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the mysqlcheck client. MySQL also includes myisamchk, a very fast command-line utility for performing these operations on MyISAM tables.
- MySQL programs can be invoked with the `-help` or `-?` option to obtain online assistance.

- **WHY USE MySQL:**

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

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

- **WHY USE JAVASCRIPT** JavaScript is one of the 3 languages all web developers must learn:

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

Example

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

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

- **OTHER USES OF JAVASCRIPT**

- Delete HTML elements
- Create new HTML elements
- Copy HTML elements
- In HTML, JavaScript is a sequence of statements that can be executed by the web browser.

- **JAVASCRIPT STATEMENTS:**

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

Semicolon;

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

- **JAVASCRIPT CODE:**

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

- **JAVASCRIPT PROPERTIES:**

- Properties are the values associated with a JavaScript object.
- A JavaScript object is a collection of unsorted properties.

- Properties can usually be changed, added, and deleted, but some are read only.

- **PHP:**

- WHAT IS PHP?**

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

- **WHAT IS PHP FILE?**

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

- **WHAT CAN PHP DO?**

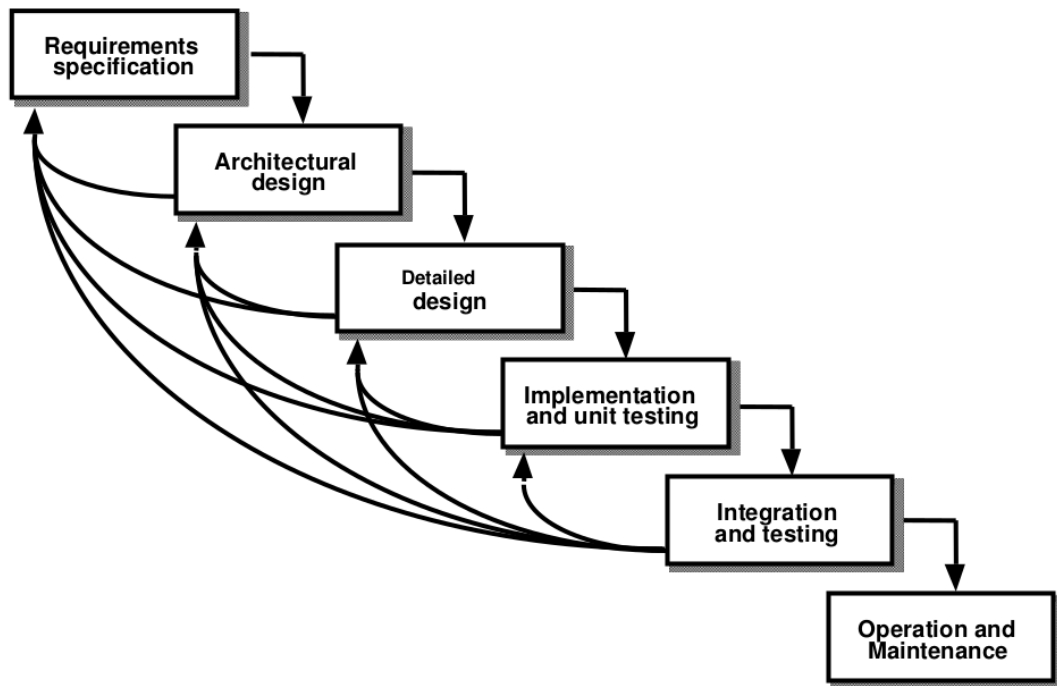
- PHP can generate dynamic page content
  - PHP can create, open, read, write, delete, and close files on the server
  - PHP can collect form data
  - PHP can send and receive cookies
  - PHP can add, delete, modify data in your database
  - PHP can restrict users to access some pages on your website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

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

- **WHY PHP?**

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

## 5 Project Methodology



This model (Iterative Model) has the same phases as the waterfall model, but with fewer restrictions. Generally the phases occur in the same order as in the waterfall model, but they may be conducted in several cycles. Useable product is released at the end of each cycle, with each release providing additional functionality.

### 5.1 Reasons For Choosing The Iterative Method

There are four factors we considered for choosing the iterative method which are:

1. Requirements

## *Based On Characteristics Of Requirements*

---

Requirements	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Are requirements easily understandable and defined?	Yes	No	No	No	No	Yes
Do we change requirements quite often?	No	Yes	No	No	Yes	No
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	No	Yes
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	Yes	No

### 2. Development Team

## *Based On Status Of Development Team*

---

Development team	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Less experience on similar projects?	No	Yes	No	No	Yes	No
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	Yes	No
Less experience on tools to be used	Yes	No	No	No	Yes	No
Availability of training if required	No	No	Yes	Yes	No	Yes

### 3. Users

## *Based On User's Participation*

---

Involvement of Users	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
User involvement in all phases	No	Yes	No	No	No	Yes
Limited user participation	Yes	No	Yes	Yes	Yes	No
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	Yes	No
Users are experts of problem domain	No	Yes	Yes	Yes	No	Yes

#### 4. Project type and associated risk



## *Based On Type Of Project With Associated Risk*

---

Project type and risk	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Project is the enhancement of the existing system	No	No	Yes	Yes	No	Yes
Funding is stable for the project	Yes	Yes	No	No	No	Yes
High reliability requirements	No	No	Yes	Yes	Yes	No
Tight project schedule	No	Yes	Yes	Yes	Yes	Yes
Use of reusable components	No	Yes	No	No	Yes	Yes
Are resources (time, money, people etc.) scarce?	No	Yes	No	No	Yes	No

## 6 DESIGN

### 6.1 System Design

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

#### 6.1.1 Introduction to Functional Decomposition Diagram(FDDs)

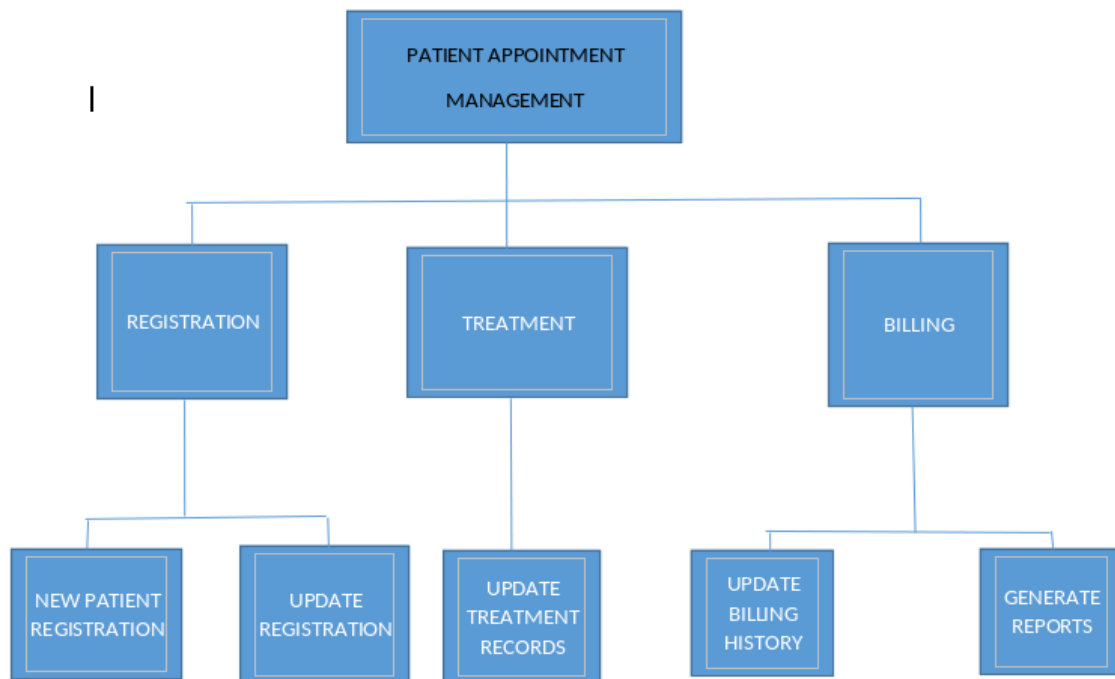
Functional decomposition diagram(FDD) is defined as a top down representation of process or function. Business analyst use functional development diagram for showing business function and break these functions into lower operations and processes. Functional decomposition diagram is used at various stages of system development.

This diagram is basically used for representing business function and operation in hierarchical manner. It is basically not used for depicting sequence of events.

The purpose of the functional decomposition diagram is to show on a single page the capabilities of an organization that are relevant to the consideration of an architecture. By examining the capabilities of an organization from a functional perspective, it is possible to quickly develop models of what the organization does without being dragged into an extended debate on how the organization does it. Once a basic functional decomposition diagram has been developed, it becomes possible to layer heat-maps on top of this diagram to show scope and decisions. For example, the capabilities to be implemented during the different phases of a change program.

#### 6.1.2 FDDs Of The Project

##### **Surgery Management Functional Decompositional Diagram**



### 6.1.3 Introduction to Data Flow Diagrams(DFDs)

Data flow diagrams (DFDs) offer a graphical technique for summarizing the movement of data between the processing steps that occur within a business process. They isolate the collections of data, or data stores, which accumulate during a process, and identify the sources of data that arise outside process boundaries. Some key characteristics of data flow diagrams are:

- **Two-dimensional summary.** DFDs offer a way to summarize the data flow characteristics of a process on a single page. As such they can provide a useful and concise summary of system-related (e.g., data-driven) process attributes.
- **Completeness.** DFDs offer a way to check the completeness of your process model, particularly as regards your understanding of the data that would be required by an information system (e.g., is all the data that would be needed for input actually available? Does each processing step produce data that could be used by subsequent steps? Is all data generated usable by an information system where necessary?). DFDs can provide a fast way to generate further questions that need to be asked about the process.
- **Processing, not processes.** DFDs refer to "process" steps. It might be more useful to think of DFD "processes" as processing steps rather than process activities. In

essence, DFDs ask one to refer to the information systems implications of any processing work that occurs during the tasks that comprise a business process. DFD terminology tends to confuse the term "process" in its connotation with business process with the term "process" that refers to a computational process executing within software (e.g., a software algorithm). Whether this represents the presumption among information engineers that everything is just a version of a computational process is a subject for further discussion at a later time (and the winner of the debate receives the Golden Nerd Award); the point here is that it is safer to think of DFD "processes" as processing steps.

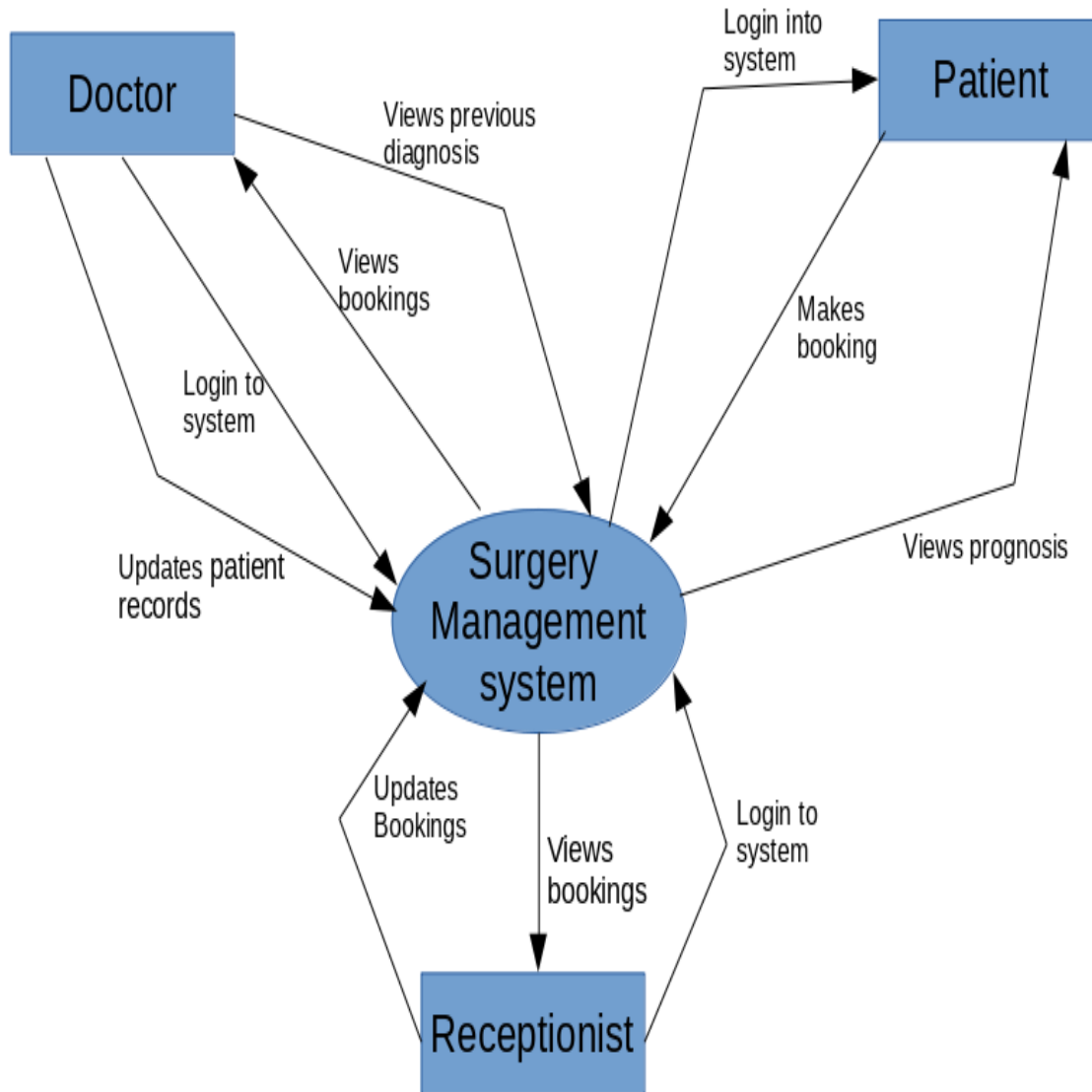
- **Patterns.** DFDs can provide a shorthand for understanding patterns that exist within the data flows supporting business processes. They can show, for example, where large amounts of data are collected, stored, transferred, generated, used, and delivered. They can highlight areas of potentially extraneous activity, and can suggest process components that do not receive the information support that they deserve (or need).

**What are some of the advantages of using DFD analysis?**

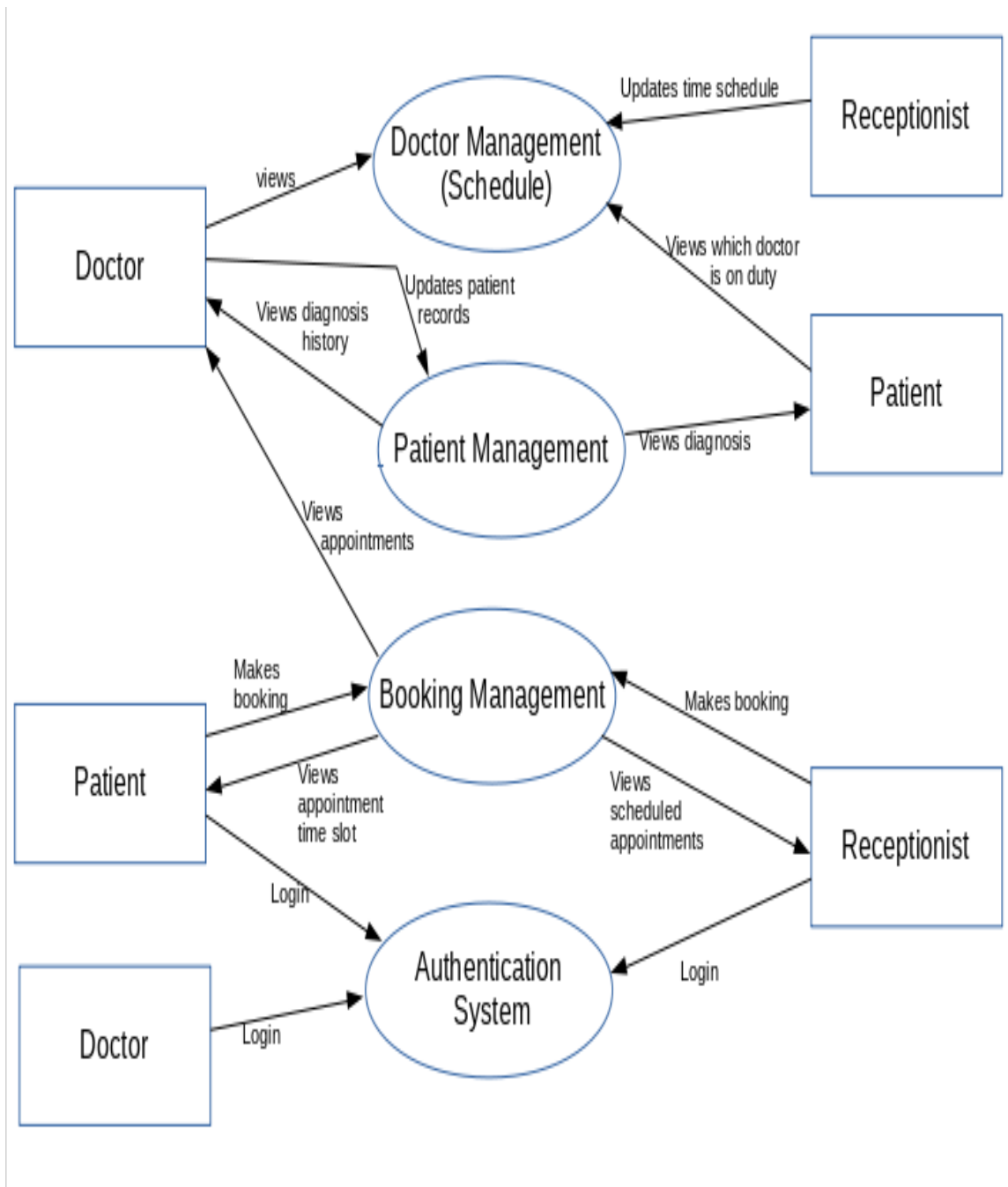
- **Data flows and process consequences.** Note how this representation of the data characteristics of banking operations enables us to start at any point in the operation (e.g., deposits, withdrawals, or bill payment), and follow the consequences of that activity through to the point where all appropriate account balances have been adjusted and reconciled. Wherever we start in the process, we can understand the processing steps that the bank would need to take to complete the relevant transaction(s) and to inform its constituents of the results.
- **Data inputs and outputs.** The DFD also makes it possible to understand what data are needed to provide appropriate inputs to any processing step. If, for example, we were to build an information system to support this individual's banking activities (in the days before Quicken and/or Microsoft Money), we would need to understand exactly what data items are represented by data flows such as "Monthly Statement", "Pay earned", "Withdraw or transfer", and other arrows shown in the diagram.
- **Simplifying complexity by isolating process components.** Note how the DFD would make it easier to capture the detail of such data flows. By isolating "Withdraw or Transfer" within the larger scheme of the banking process, the DFD makes it possible to consider the details of the data items included in this flow without reference to the flows affecting other processing steps. All of the flows affecting withdrawals (e.g., processing step 3.0, "Withdraw funds from account") are isolated as entering or leaving processing step 3.0. At the time that DFDs were developed, this shift towards modularizing data flows and processing elements represented a major step forward in enabling systems analysts to add useful structure to process representations rapidly and easily.

#### 6.1.4 FDDs Of The Project

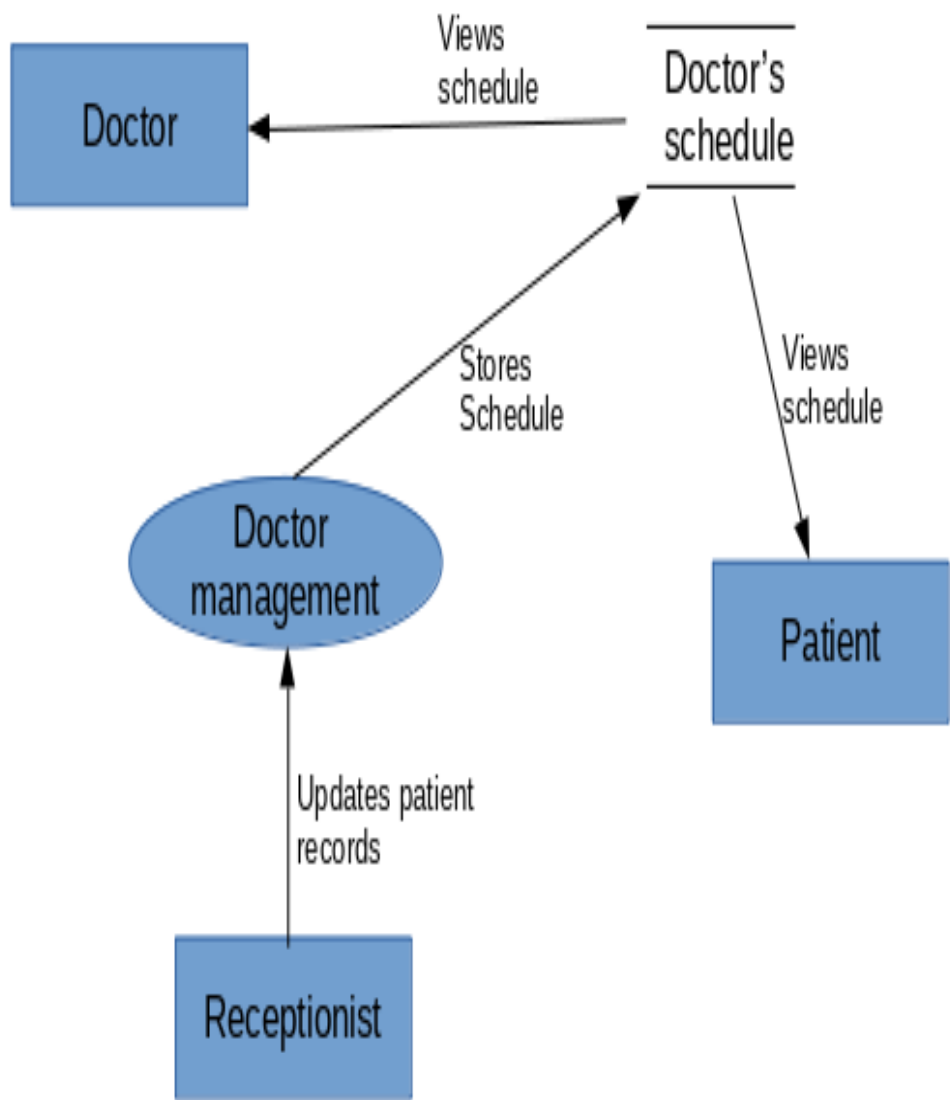
##### Surgery Level 0(Context) Data Flow Diagram

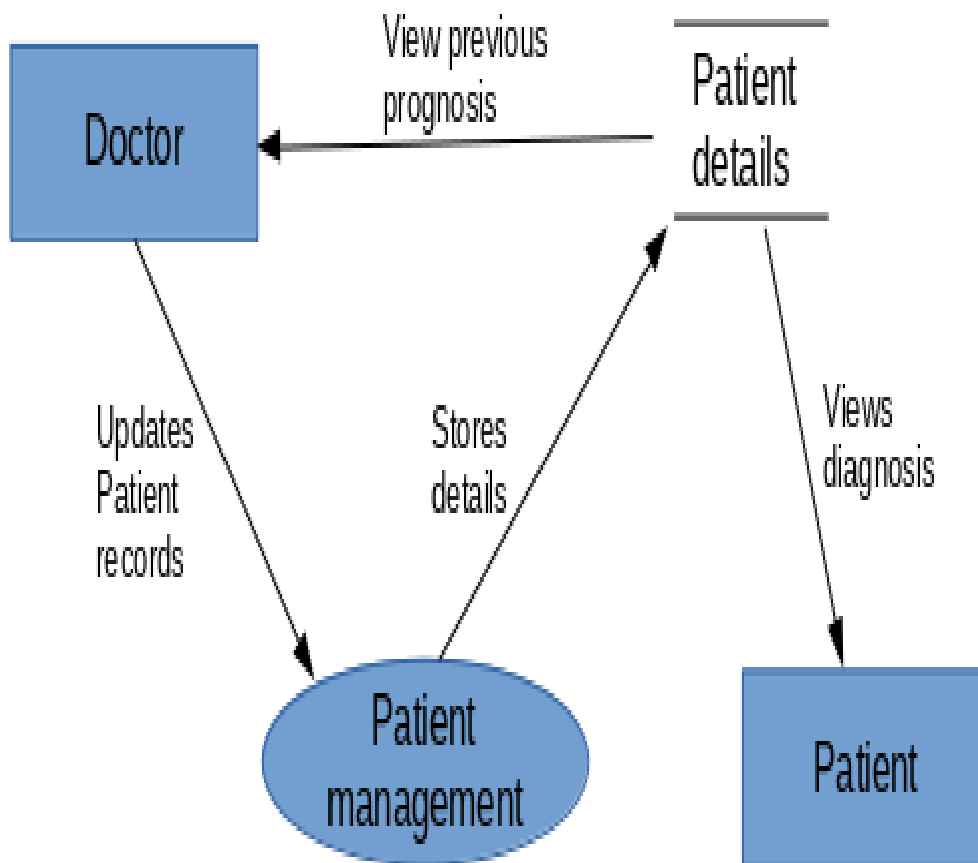


## Surgery Level 1 Data Flow Diagram

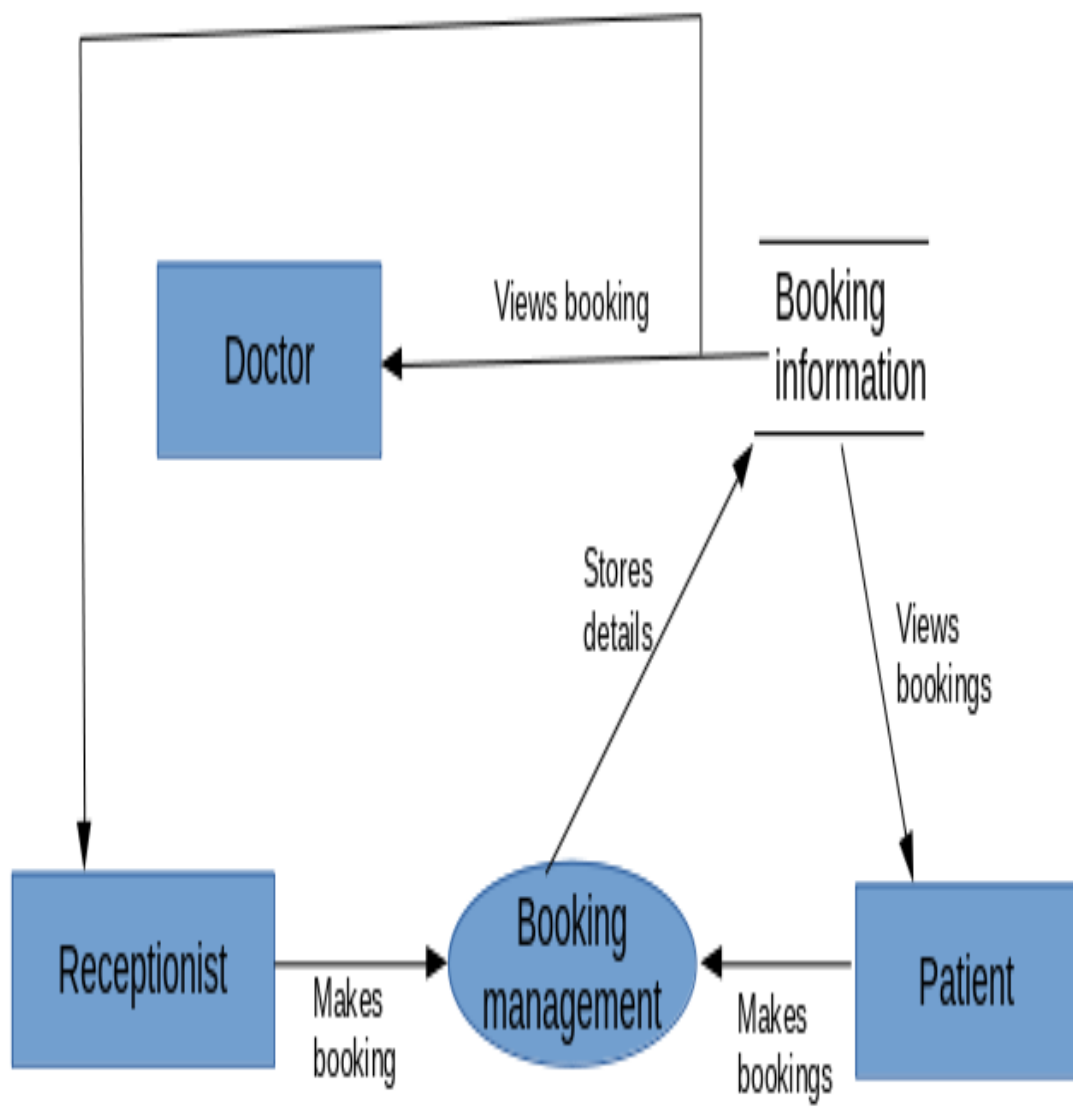


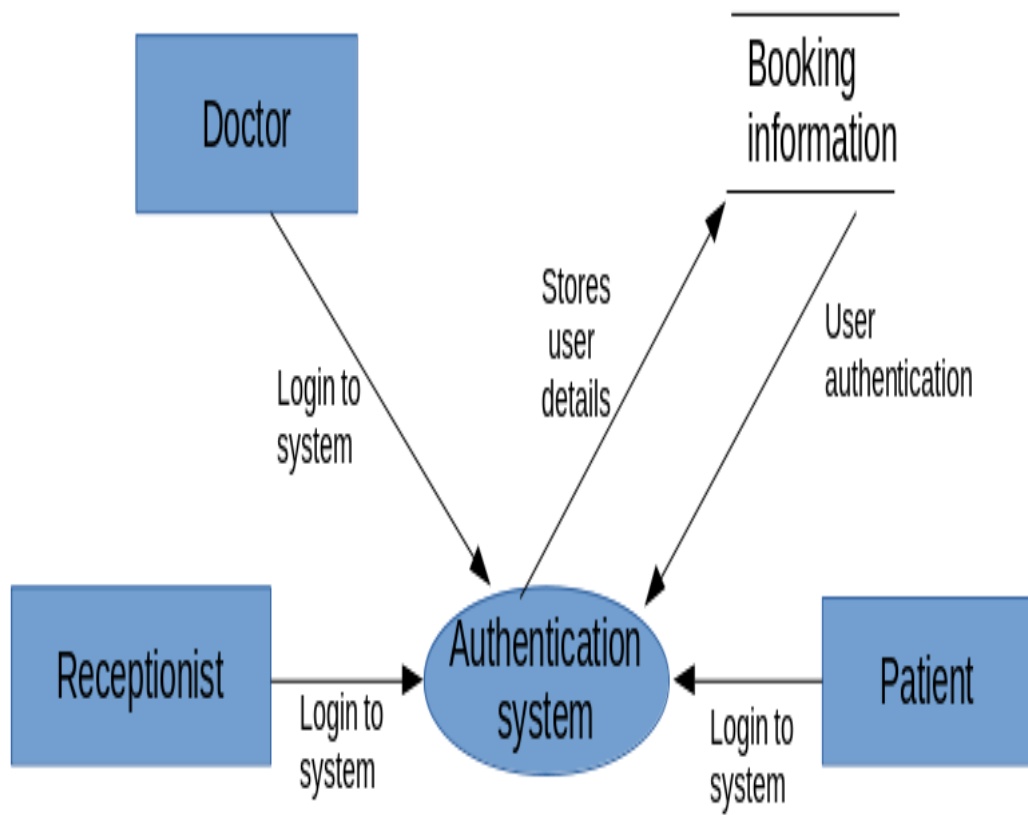
# Surgery Level 2 Data Flow Diagram











### 6.1.5 Introduction to Entity-Relationship(ER) Diagrams

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other.

#### ER Diagram with basic objects

In the diagram, the elements inside rectangles are called entities while the items inside diamonds denote the relationships between entities.

#### ER Diagrams Usage

While able to describe just about any system, ER diagrams are most often associated with complex databases that are used in software engineering and IT networks. In particular, ER diagrams are frequently used during the design stage of a development process in order to identify different system elements and their relationships with each other. For example, an inventory software used in a retail shop will have a database that monitors elements such as purchases, item, item type, item source and item price.

#### ER diagram example with entity having attributes

In the diagram, the information inside the oval shapes are attributes of a particular entity.

**History of ER Diagrams** ER diagrams are visual tools that are used in the Entity-Relationship model initially proposed by Peter Chen in 1976 to create a uniform convention that considers both relational database and network views. Chen envisioned the ER model as a conceptual modeling approach that views real world data as systems of entities and relationships. Entities are data objects that maintain different relationships with each other. Additionally, entities are also described further using attributes.

Since 1976, the ER model has been expanded and is sometimes used in business management, product development, and strategy formulations. However, database design remains its primary application.

#### Elements in ER diagrams

There are three basic elements in an ER Diagram: entity, attribute, relationship. There are more elements which are based on the main elements. They are weak entity, multivalued attribute, derived attribute, weak relationship and recursive relationship. Cardinality and ordinality are two other notations used in ER diagrams to further define relationships.

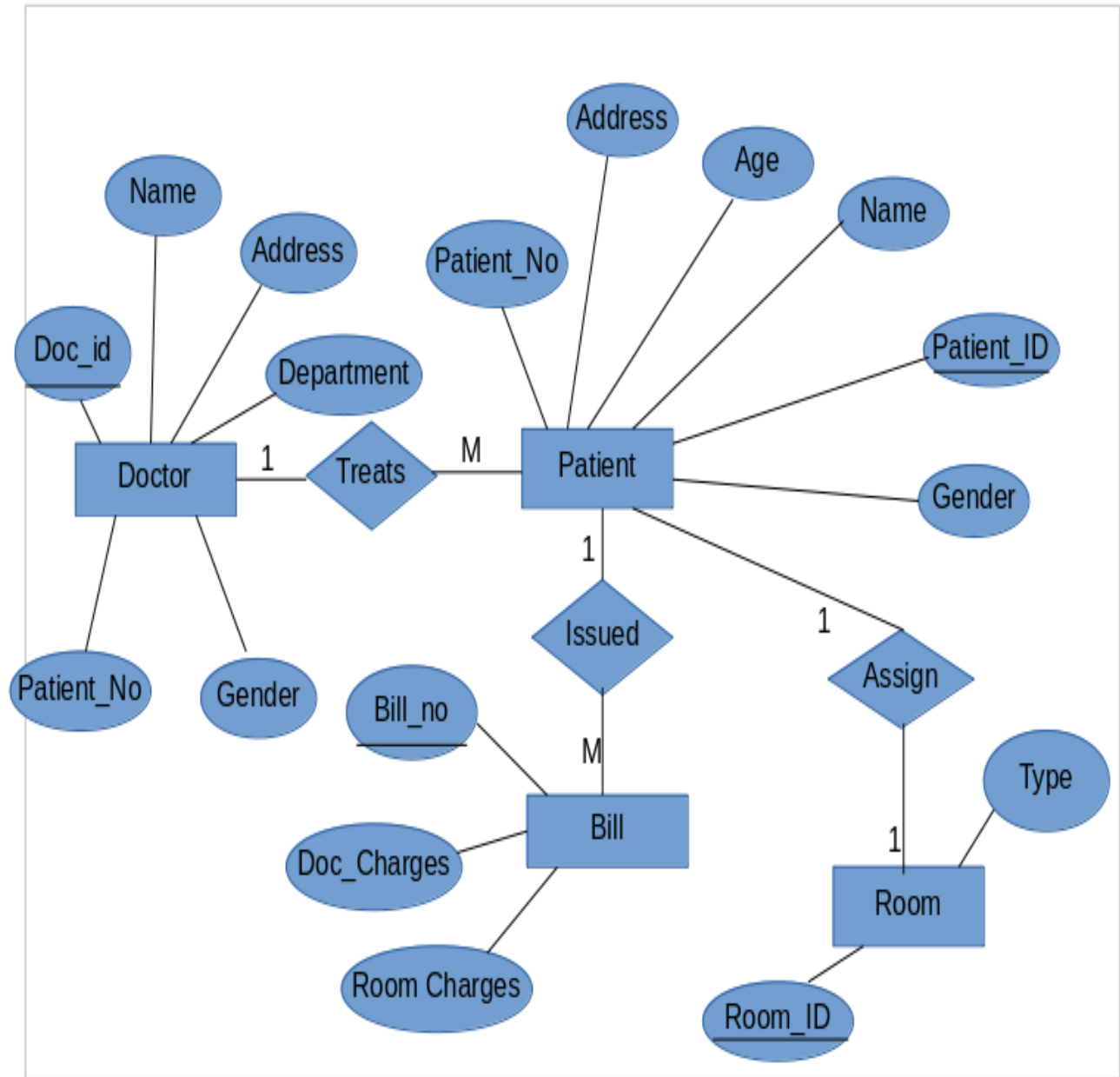
- **Entity** An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses,

subjects, fees, and other items. Entities are represented in ER diagrams by a rectangle and named using singular nouns.

- **Weak Entity** A weak entity is an entity that depends on the existence of another entity. In more technical terms it can be defined as an entity that cannot be identified by its own attributes. It uses a foreign key combined with its attributed to form the primary key. An entity like order item is a good example for this. The order item will be meaningless without an order so it depends on the existence of order.
- **Attribute** An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. For example, the attribute Inventory Item Name is an attribute of the entity Inventory Item. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes. For example, the attribute "customer address" can have the attributes number, street, city, and state. These are called composite attributes. Note that some top level ER diagrams do not show attributes for the sake of simplicity. In those that do, however, attributes are represented by oval shapes
- **Multivariate Attribute** If an attribute can have more than one value it is called an multivariate attribute. It is important to note that this is different to an attribute having its own attributes. For example a teacher entity can have multiple subject values.
- **Derived Attribute** An attribute based on another attribute. This is found rarely in ER diagrams. For example for a circle the area can be derived from the radius.
- **Relationship** A relationship describes how entities interact. For example, the entity "carpenter" may be related to the entity "table" by the relationship "builds" or "makes". Relationships are represented by diamond shapes and are labeled using verbs.
- **Recursive Relationship** If the same entity participates more than once in a relationship it is known as a recursive relationship. In the below example an employee can be a supervisor and be supervised, so there is a recursive relationship.
- **Cardinality and Ordinality** These two further defines relationships between entities by placing the relationship in the context of numbers. In an email system, for example, one account can have multiple contacts. The relationship in this case follows a "one to many" model. There are number of notations used to present cardinality in ER diagrams. Chen, UML, Crows foot, Bachman are some of the popular notations. Creately supports Chen, UML and Crows foot notations. The following example uses UML to show cardinality.

### 6.1.6 Entity-Relationship Diagram for The Project

#### ER Diagram for the Surgery



## 7 SYSTEM IMPLEMENTATION

### 7.1 Introduction

We implemented the system using web technologies such as HTML, Javascript and CSS because it is very paramount for the system to be online for easier access. This enables the users (doctor, patients and other members of staff) to remotely access the system. This means that the patient can make reservations from anywhere. System maintenance can be performed from any place with internet connectivity and therefore the support team does not have to be physically located within the premises of the Surgery.

The required system is simple and does not require a lot of expertise to realise. The databases are created using the MySQL relational database management system. MySQL relational database management system. MySQL is an open source RDBM, which is based on SQL (structured query language). SQL is used for adding, removing, modifying information in the database. Standard SQL commands such as ADD, DROP, INSERT and UPDATE are used in MySQL. This system is database driven and as such uses PHP to access information from the database.

### 7.2 Sample code

#### 7.2.1 Add Doctor PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,
initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/
4.0.0-beta/css/bootstrap.min.css"

integrity="sha384-Y6pD6FV/Vv2HJnA6t+vslU6fwYXj
CFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"

crossorigin="anonymous">
```

```

</head>
<body style="background-color:#3498DB
; color:white;padding-top:100px;
text-align:center;">
<h3>Your doctor has been added.</h3><br><br>
<a href="admin-panel.php"
class="btn btn-outline-light">Go to Admin Panel</a>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/
jquery-3.2.1.slim.min.js"

integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/
rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/
ajax/libs/popper.js/1.11.0/umd/popper.min.js"

integrity="sha384-b/U6ypiBEHpOf/
4+1nzFpr53nxSS+GLCKfwBdFNTxtclqqenISfwAzpKaMNFNmj4"

crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/js/bootstrap.min.js"

integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3S
EJAg45jXxnGIfYzk4Si90RDIqNm1"

crossorigin="anonymous"></script>
</body>
</html>

```

### 7.2.2 Admin Panel PHP

```

<?php
include('func.php');
if(!isset($_SESSION['username']))
    echo "session expired";
else
    display_admin_panel();

```

### 7.2.3 Appointment PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,
initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/css/bootstrap.min.css"

integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U6
fwYXjCFtcEpHbNJ0lyAFsXTsJBbfaDjzALeQsN6M"

crossorigin="anonymous">
</head>
<body style="background-color:#3498DB;
color:white;padding-top:100px;
text-align:center;">

<h3>Your appointment has been booked.</h3><br><br>

<a href="admin-panel.php"
class="btn btn-outline-light">Return to admin panel</a>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/
jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/
rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/
ajax/libs/popper.js/1.11.0/umd/popper.min.js"

integrity="sha384-b/U6ypiBEHpOf/
4+1nzFpr53nxSS+GLCKfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
```



```

crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/js/bootstrap.min.js"

integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsH
bi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1"

crossorigin="anonymous"></script>
</body>
</html>
<?php
include("func.php");
if(isset($_POST['entry_submit'])) {
$fname=$_POST['fname'];
$lname=$_POST['lname'];
$email=$_POST['email'];
$contact=$_POST['contact'];
$doctor=$_POST['doctor'];
$payment=$_POST['payment'];

$query="insert into appointmenttb(
fname,lname,email,contact,doctor,payment)
values(' $fname ',' $lname ',' $email ',
' $contact ',' $doctor ',' $payment ');";

$result=mysqli_query($con,$query);

if($result)
header("Location:appointment.php");
}
?>

```

#### 7.2.4 Search PHP

```

<?php
$con=mysqli_connect("localhost","root","","hmsdb");
if(isset($_POST['search_submit'])) {
$contact=$_POST['contact'];
$query="select *
from appointmenttb where contact=' $contact '";
$result=mysqli_query($con,$query);

```

```

echo ' <!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width , initial-scale=1,
shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href=
"https://maxcdn.bootstrapcdn.com/bootstrap/
4.0.0-beta/css/bootstrap.min.css"
integrity=
"sha384-Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEp
HbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<body style="background-color:#3498DB;
color:white;text-align:center;
padding-top:50px;" >
<div class="container"
style="text-align:left;" >
<center><h3>Search Results</h3></center><br>
<table class="table table-hover">
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Contact</th>
<th>Doctor</th>
<th>Payment</th>
</tr>
</thead>
<tbody>
';
while ($row=mysqli_fetch_array ($result)){
$name=$row[ 'fname' ];
$lname=$row[ 'lname' ];
$email=$row[ 'email' ];

```

```

$contact=$row[ 'contact' ];
$doctor=$row[ 'doctor' ];
$payment=$row[ 'payment' ];
echo '<tr>
<td>'. $fname. '</td>
<td>'. $lname. '</td>
<td>'. $email. '</td>
<td>'. $contact. '</td>
<td>'. $doctor. '</td>
<td>'. $payment. '</td>
</tr>';
}
echo '</tbody></table></div>
<!--_Optional_JavaScript_-->
<!--_jQuery_first_,_then_Popper.js_,
then_Bootstrap_JS_-->
<script_src=
"https://code.jquery.com/
jquery-3.2.1.slim.min.js"
integrity=
"sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/
rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin=
"anonymous"></script>
<script_src=
"https://cdnjs.cloudflare.com/ajax/libs/
popper.js/1.11.0/umd/popper.min.js"
integrity=
"sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GL
CkfwBdFNTxtclqgenISfwAzpKaMNFNmj4"
crossorigin=
"anonymous"></script>
<script_src=
"https://maxcdn.bootstrapcdn.com/bootstrap/
4.0.0-beta/js/bootstrap.min.js"
integrity=
"sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO
3SEJAg45jXxnGIfYzk4Si90RDIqNm1"
crossorigin=
"anonymous"></script>
</body>
</html>';

```

```
}  
?>
```

### 7.2.5 Update PHP

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<!-- Required meta tags -->  
<meta charset="utf-8">  
<meta name="viewport"  
content="width=device-width,  
initial-scale=1, shrink-to-fit=no">  
  
<!-- Bootstrap CSS -->  
<link rel="stylesheet"  
  
href="https://maxcdn.bootstrapcdn.com/  
bootstrap/4.0.0-beta/css/bootstrap.min.css"  
  
integrity="sha384-Y6pD6FV/Vv2HJnA6t+vslU6  
fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"  
  
crossorigin="anonymous">  
</head>  
<body style="background-color:#3498DB;  
color:white;padding-top:100px;  
text-align:center;">  
<h3>Your payment status have been updated.  
</h3><br><br>  
<a href="admin-panel.php"  
class="btn btn-outline-light">Go Back</a>  
<!-- Optional JavaScript -->  
<!-- jQuery first, then Popper.js, then Bootstrap JS -->  
<script src="https://code.jquery.com/  
jquery-3.2.1.slim.min.js"  
  
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/  
rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"  
  
crossorigin="anonymous"></script>  
<script src="https://cdnjs.cloudflare.com/  
ajax/libs/popper.js/1.11.0/umd/popper.min.js"
```

```
integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53n
xSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
```

```
crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/js/bootstrap.min.js"
```

```
integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi
047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1"
```

```
crossorigin="anonymous"></script>
</body>
</html>
```

## 7.2.6 Logout PHP

```
<?php
session_start();
session_destroy();
?>
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,
initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/css/bootstrap.min.css"

integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs
lU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"

crossorigin="anonymous">
</head>
<body style="background-color:#3498DB;
color:white;padding-top:100px;
text-align:center;">
```

```

<h3>You have logged out.</h3><br><br>
<a href="index.php"
  class="btn btn-outline-light">Try Again</a>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/
jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr
/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/
ajax/libs/popper.js/1.11.0/umd/popper.min.js"

integrity="sha384-b/U6ypiBEHpOf/
4+1nzFpr53nxSS+GLCKfwBdFNTxtclqqenISfwAzpKaMNFNmj4"

crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/js/bootstrap.min.js"

integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi
047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1"

crossorigin="anonymous"></script>
</body>
</html>

```

### 7.2.7 Error PHP

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width,
      initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->

```

```

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/css/bootstrap.min.css"

integrity="sha384-Y6pD6FV/Vv2HJnA6t+vslU6
fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"

crossorigin="anonymous">
</head>
<body style="background-color:#3498DB;
color:white;padding-top:100px;
text-align:center;">

<h3>Your username or password was
incorrect.<br>Please try again.</h3><br><br>

<a href="index.php"
class="btn btn-outline-light">Try Again</a>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/
jquery-3.2.1.slim.min.js"

integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr
/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/
ajax/libs/popper.js/1.11.0/umd/popper.min.js"

integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxS
S+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4"

crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/js/bootstrap.min.js"

integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi
047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIQNm1"

```

```

        crossorigin="anonymous"></script>
</body>
</html>

```

## 7.2.8 Surgery Management Database System SQL

```

— phpMyAdmin SQL Dump
— version 4.6.4
— https://www.phpmyadmin.net/
—
— Host: 127.0.0.1
— Generation Time: Oct 31, 2017 at 03:15 PM
— Server version: 5.7.14
— PHP Version: 5.6.25

```

```

SET SQLMODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

```

```

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

```

```

—
— Database: 'hmsdb'
—

```

---

```

—
— Table structure for table 'appointmenttb'
—

```

```

CREATE TABLE 'appointmenttb' (
  'fname' varchar(80) NOT NULL,
  'lname' varchar(80) NOT NULL,
  'email' varchar(80) NOT NULL,
  'contact' varchar(80) NOT NULL,
  'doctor' varchar(80) NOT NULL,
  'payment' varchar(40) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```



```

—
— Dumping data for table 'appointmenttb'
—

INSERT INTO 'appointmenttb' ('fname', 'lname', 'email',
'contact', 'doctor', 'payment') VALUES ('jdsnk', 'kjndk',
'jkdnk', 'kjdfn', 'Dr._Punam_Shaw', ''), ('dsskj',
'dkjn', 'dnkjn', 'kjdn', 'Dr._Ashok_Goyal', ''), ('sdkjn',
'dskjnk', 'kfdjnkjn', 'jndkjn', 'Dr._Ashok_Goyal',
'Pay_later'), ('provat', 'dutta', 'pd@gmail.com',
'9474764197', 'Dr._Ashok_Goyal', 'Pay_later'),
('sample1', 'sample1', 'sample1@gmail.com', '1982272721',
'Dr._Punam_Shaw', 'Paid'), ('Prithwiraj', 'Dutta',
'duttaprichwiraj9@gmail.com', '+917586827481',
'Dr._Ashok_Goyal', 'Pay_later'), ('rohit', 'sharma',
'rs@yahoo.com', '2838393923', 'Dr._Punam_Shaw', 'Paid'),
('dexter', 'dutta', 'dd@yahoo.com', '327262728',
'Dr._Ashok_Goyal', 'paid');

```

---

```

— Table structure for table 'doctb'

```

```

CREATE TABLE 'doctb' (
'name' varchar(50) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

—
— Dumping data for table 'doctb'
—

```

```

INSERT INTO 'doctb' ('name') VALUES
('Dr._Ashok_Kumar'),
('Dr._Pravin_Malotra'),
('Dr._Prithwiraj_Dutta'),
('Dr._Rohit_Mehta');

```

---

```

—
— Table structure for table 'logintb'
—

```

```
CREATE TABLE 'logintb' (
  'username' varchar(80) NOT NULL,
  'password' varchar(80) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
—
— Dumping data for table 'logintb'
—
```

```
INSERT INTO 'logintb' ('username',
  'password') VALUES
  ('admin', 'admin123');
```

```
/*!40101 SET CHARACTER_SET_CLIENT=
@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=
@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=
@OLD_COLLATION_CONNECTION */;
```

## 7.2.9 Func PHP

```
<?php
session_start();
$con=mysqli_connect("localhost","root","","hmsdb");
if(isset($_POST['login_submit'])){
$username=$_POST['username'];
$password=$_POST['password'];
$query="select * from logintb where
username='$username' and password='$password'";
$result=mysqli_query($con,$query);
if(mysqli_num_rows($result)==1)
{
$_SESSION['username']=$username;
header("Location:admin-panel.php");
}
else
header("Location:error.php");
}
if(isset($_POST['update_data']))
{
$contact=$_POST['contact'];
$status=$_POST['status'];
```

```

$query="update appointmenttb set
payment='$status' where contact='$contact'";
$result=mysqli_query($con,$query);
if($result)
header("Location: updated.php");
}
function display_docs()
{
global $con;
$query="select * from doctb";
$result=mysqli_query($con,$query);
while($row=mysqli_fetch_array($result))
{
$name=$row['name'];
echo '<option
value="'. $name. '">'. $name. '</option>';
}
}
if(isset($_POST['doc_sub']))
{
$name=$_POST['name'];
$query="insert into doctb(name) values('$name)";
$result=mysqli_query($con,$query);
if($result)
header("Location: adddoc.php");
}
function display_admin_panel(){
global $con;
echo '<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,
initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet"
type="text/css" href="font-awesome-4.7.0/
css/font-awesome.min.css">

<link rel="stylesheet" href="style.css">

```

```

<!--_Bootstrap_CSS_-->
<link_rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/css/bootstrap.min.css"

integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vs1U
6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"

crossorigin="anonymous">

<nav_class="navbar navbar-expand-lg
navbar-dark bg-primary fixed-top">

<a_class="navbar-brand" _href="#">
<i_class="fa fa-user-plus"
aria-hidden="true">
</i>_Global_Hospital</a>
<button_class="navbar-toggler"
type="button" _data-toggle="collapse"
data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent"
aria-expanded="false"
aria-label="Toggle_navigation">
<span_class="navbar-toggler-icon"></span>
</button>

<div_class="collapse navbar-collapse"
id="navbarSupportedContent">
<ul_class="navbar-nav _mr-auto">
<li_class="nav-item">
<a_class="nav-link"
href="logout.php"><i_class="fa
fa-sign-out" _aria-hidden="true"></i>Logout</a>
</li>
<li_class="nav-item">
<a_class="nav-link" _href="#"></a>
</li>
</ul>
<form_class="form-inline _my-2 _my-lg-0"
method="post" _action="search.php">
<input_class="form-control _mr-sm-2" _type="text"

```

```

placeholder="enter_contact_number"
aria-label="Search" _name="contact">
<input _type="submit"
class="btn btn-outline-light _my-2 _my-sm-0 btn
btn-outline-light" _id="inputbtn"
name="search_submit" _value="Search">
</form>
</div>
</nav>
</head>
<style _type="text/css">
button: hover { cursor: pointer; }
#inputbtn: hover { cursor: pointer; }
</style>
<body _style="padding-top: 50px;" >
<div _class="jumbotron" _id="ab1"></div>
<div _class="container-fluid"
style="margin-top: 50px;" >
<div _class="row">
<div _class="col-md-4">
<div _class="list-group" _id="list-tab"
role="tablist">
<a _class="list-group-item list-group-item-action
active" _id="list-home-list"
data-toggle="list" _href="#list-home" _role="tab"
aria-controls="home">Appointment</a>
<a _class="list-group-item list-group-item-action"
id="list-profile-list"
data-toggle="list" _href="#list-profile" _role="tab"
aria-controls="profile">Payment_Status</a>
<a _class="list-group-item list-group-item-action"
id="list-messages-list"
data-toggle="list" _href="#list-messages" _role="tab"
aria-controls="messages">Prescription</a>
<a _class="list-group-item list-group-item-action"
id="list-settings-list"
data-toggle="list" _href="#list-settings" _role="tab"
aria-controls="settings">Doctors_Section</a>
<a _class="list-group-item list-group-item-action"
id="list-attend-list"
data-toggle="list" _href="#list-attend" _role="tab"
aria-controls="settings">Attendance</a>

```

```

</div><br>
</div>
<div class="col-md-8">
<div class="tab-content" id="nav-tabContent">
<div class="tab-pane fade show active" id="list-home"
role="tabpanel" aria-labelledby="list-home-list">
<div class="container-fluid">
<div class="card">
<div class="card-body">
<center><h4>Create an appointment</h4></center><br>
<form class="form-group" method="post" action="appointment.php">
<div class="row">
<div class="col-md-4"><label>First Name:</label></div>
<div class="col-md-8"><input type="text" class="form-control"
name="fname"></div><br><br>
<div class="col-md-4"><label>Last Name:</label></div>
<div class="col-md-8"><input type="text" class="form-control"
name="lname"></div><br><br>
<div class="col-md-4"><label>Email id:</label></div>
<div class="col-md-8"><input type="text" class="form-control"
name="email"></div><br><br>
<div class="col-md-4"><label>Contact Number:</label></div>
<div class="col-md-8"><input type="text" class="form-control"
name="contact"></div><br><br>
<div class="col-md-4"><label>Doctor:</label></div>
<div class="col-md-8">
<select name="doctor" class="form-control" >' ;

$cmd = $con->query("SELECT * FROM doctb") or die($con->error);
while ($row = $cmd->fetch_assoc()) {
echo '<option name="'. $row['id']. '">'. $row['name']. '</option>';

}

echo '
<!--<option value="Doctor1">Doctor1</option>-->
<?php display_docs();?>
</select>
</div><br><br>
<div class="col-md-4"><label>Payment:</label></div>
<div class="col-md-8">
<select name="payment" class="form-control" >

```

```

<option _value="Paid">Paid</option>
<option _value="Pay_later">Pay_later </option>
</select>
</div><br><br><br>
<div _class="col-md-4">
<input _type="submit" _name="entry_submit" _value="Create_new_entry"
class="btn btn-primary" _id="inputbtn">
</div>
<div _class="col-md-8"></div>
</div>
</form>
</div>
</div>
</div><br>
</div>
<div _class="tab-pane fade" _id="list-profile" _role="tabpanel"
aria-labelledby="list-profile-list">
<div _class="card">
<div _class="card-body">
<form _class="form-group" _method="post" _action="func.php">
<input _type="text" _name="contact" _class="form-control"
placeholder="enter_contact"><br>
<select _name="status" _class="form-control">
<option _value="paid">paid</option>
<option _value="pay_later">pay_later </option>
</select><br><hr>
<input _type="submit" _value="update" _name="update_data"
class="btn btn-primary">
</form>
</div>
</div><br><br>
</div>
<div _class="tab-pane fade" _id="list-messages" _role="tabpanel"
aria-labelledby="list-messages-list">...</div>
<div _class="tab-pane fade" _id="list-settings" _role="tabpanel"
aria-labelledby="list-settings-list">
<form _class="form-group" _method="post" _action="func.php">
<label>Doctors _name:</label>
<input _type="text" _name="name" _placeholder="enter_doctors
name" _class="form-control">
<br>
<input _type="submit" _name="doc_sub" _value="Add_Doctor"

```

```

class="btn btn-primary">
</form>
</div>
<div class="tab-pane fade" id="list-attend" role="tabpanel"
aria-labelledby="list-attend-list">...</div>
</div>
</div>
</div>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/
jquery-3.2.1.slim.min.js"

integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KChRr
/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/
libs/popper.js/1.11.0/umd/popper.min.js"

integrity="sha384-b/U6ypiBEHpOf/
4+1nzFpr53nxSS+GLCKfwBdFNTxtclqqenISfwAzpKaMNFNmj4"

crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com
/bootstrap/4.0.0-beta/js/bootstrap.min.js"
integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047N
rKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1"
crossorigin="anonymous"></script>
<!-- Sweet alert.js -->
<script src="
https://cdnjs.cloudflare.com/ajax/libs/
limonte-sweetalert2/6.10.1/
sweetalert2.all.min.js"></script>
</body>
</html>';
}
?>

```

## 7.2.10 STYLE CSS



```
#ab1{
    border-radius:0;background:url("images/head.jpeg");
    background-size:cover;height:400px;
}
```

### 7.2.11 Index PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,
initial-scale=1,shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/css/bootstrap.min.css"

integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6
fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M"
crossorigin="anonymous">
</head>
<style type="text/css">
#inputbtn:hover{cursor:pointer;}
</style>
<body style="background:url('images/backgroundimg.jpeg');
background-size:cover;">
<div class="container-fluid"
style="margin-top:60px;margin-bottom:60px;color:#34495E;">
<div class="row">
<div class="col-md-1"></div>
<div class="col-md-4">
<div class="card">

<div class="card-body">
<center>
<h5>Admin Login</h5><br>
<form class="form-group" method="post" action="func.php">
<div class="row">
<div class="col-md-4"><label>Username: </label></div>
```

```

<div class="col-md-8"><input type="text" name="username"
class="form-control" placeholder="enter_username"
required/></div><br><br>
<div class="col-md-4"><label>Password: </label></div>
<div class="col-md-8"><input type="password"
class="form-control" name="password" placeholder="enter_password"
required/></div><br><br><br>
</div>
<center><input type="submit" id="inputbtn"
name="login_submit" value="Login" class="btn btn-primary"></center>
</form>
</center>
</div>
</div>
</div>
<div class="col-md-7"></div>
</div>
</div>

```

```

<!-- Optional JavaScript -->
<!-- jQuery first , then Popper.js , then Bootstrap JS -->
<script src="https://code.jquery.com/
jquery-3.2.1.slim.min.js"

```

```

integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/
rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

```

```

crossorigin="anonymous"></script>

```

```

<script src="https://cdnjs.cloudflare.com/
ajax/libs/popper.js/1.11.0/umd/popper.min.js"

```

```

integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCKfwBdF
NTxtclqgenISfwAzpKaMNFNmj4"

```

```

crossorigin="anonymous"></script>

```

```

<script src="https://maxcdn.bootstrapcdn.com/
bootstrap/4.0.0-beta/js/bootstrap.min.js"

```

```

integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047Nr
KGLO3SEJAg45jXnGIfYzk4Si90RDIqNm1"

```

```
crossorigin="anonymous"></script>  
</body>  
</html>
```

## 8 TESTING

### 8.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 8.2 Testing Methods

#### TYPES OF TESTING:

- **Unit testing:** Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.
- **Integration testing:** Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.
- **Functional test:** Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

- **System Test:** System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.
- **White Box Testing:** System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.
- **Black Box Testing:** Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works.
- **Unit Testing:** Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.
- **Test strategy and approach:** Field testing will be performed manually and functional tests will be written in detail.
- **Test objectives** The entry screen, messages and responses must not be delayed.
  - All field entries must work properly.
  - Pages must be activated from the identified link.
  - The entry screen, messages and responses must not be delayed.
- **Features to be tested**

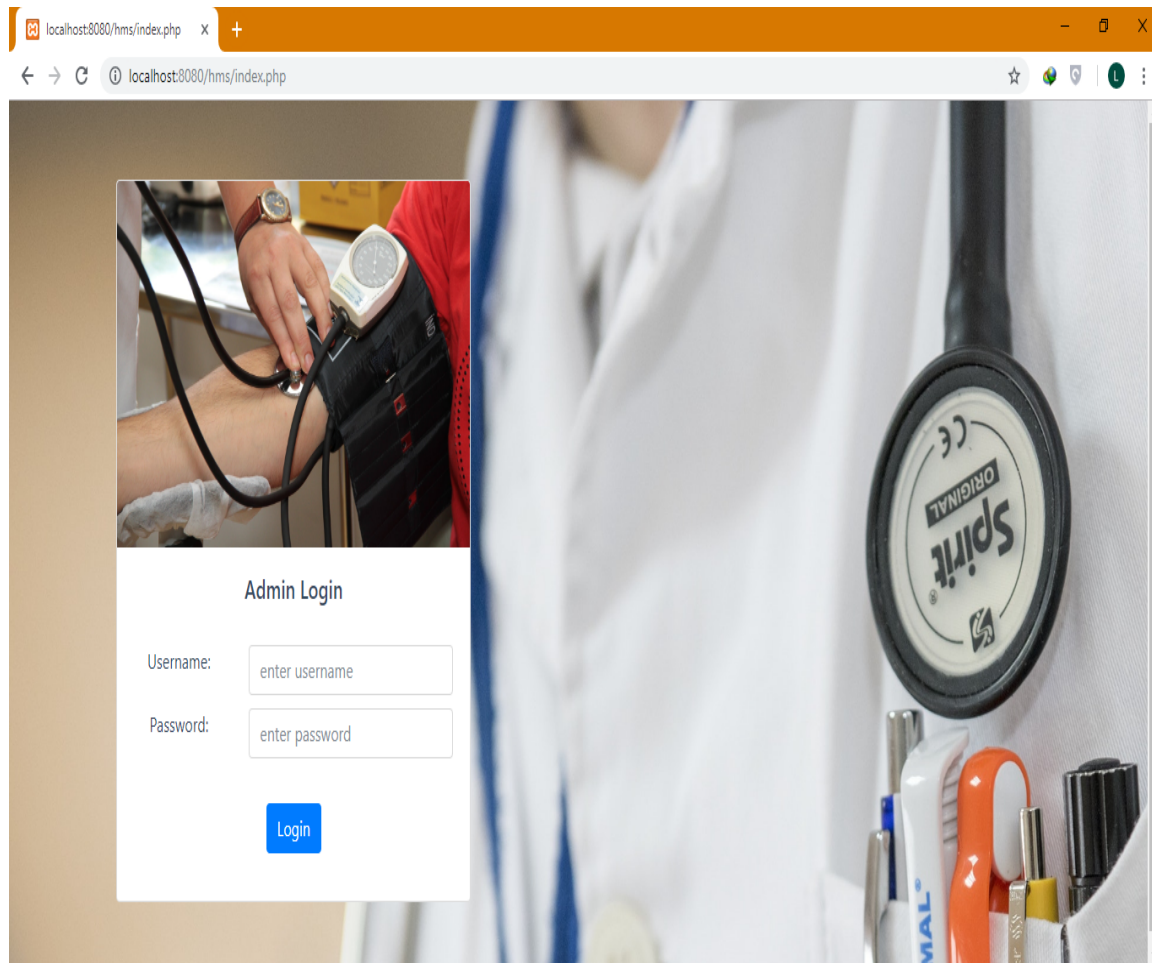
- Verify that the entries are of the correct format
  - No duplicate entries should be allowed
  - All links should take the user to the correct page.
- **Integration Testing:** Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

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

- **Test Results:** All the test cases mentioned above passed successfully. No defects encountered.
- **Acceptance Testing:** User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements .
- **Test Results:** All the test cases mentioned above passed successfully. No defects encountered .

## 9 SAMPLE SCREENSHOTS

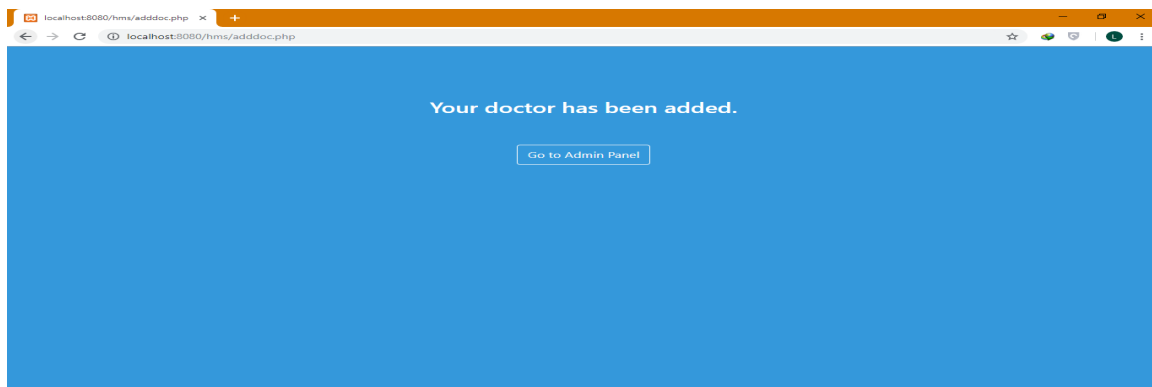
### 9.1 Login



## 9.2 Appointment

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/hms/admin-panel.php'. The page has a blue header with the 'Global Hospital' logo and a 'Logout' link. A search bar with the placeholder 'enter contact number' and a 'Search' button is also present. The main content area is divided into two sections. On the left is a sidebar menu with the following items: 'Appointment' (highlighted in blue), 'Payment Status', 'Prescription', 'Doctors Section', and 'Attendance'. On the right is a form titled 'Create an appointment'. The form contains the following fields: 'First Name' (text input with value 'Lorthar'), 'Last Name' (text input with value 'Mabandla'), 'Email id' (text input with value 'lortharmatheus@gmail.com'), 'Contact Number' (text input with value '+263776721164'), 'Doctor' (dropdown menu with value 'Doctor1'), and 'Payment' (dropdown menu with value 'Paid'). At the bottom of the form is a blue button labeled 'Create new entry'.

## 9.3 New Doctor





## 9.4 Payment Status

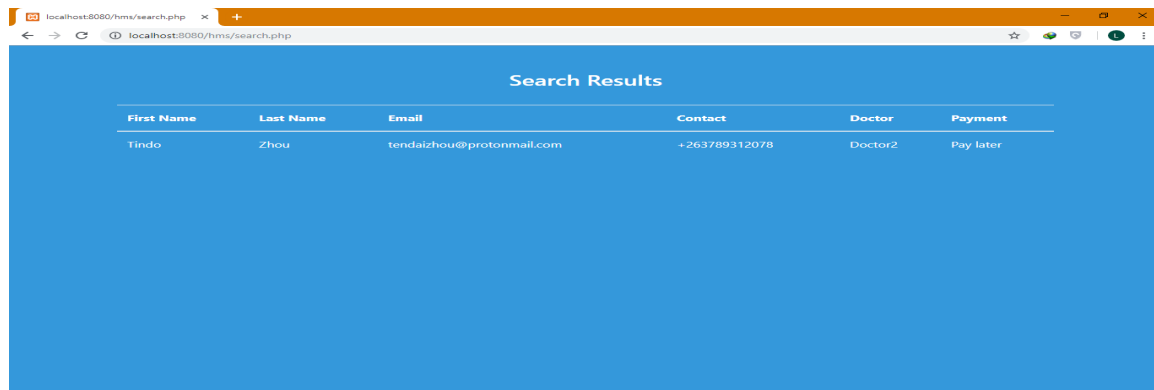
The screenshot shows a web browser window with the URL `localhost:8080/hms/admin-panel.php`. The page has a blue header with the "Global Hospital" logo and a "Logout" link. A search bar with the placeholder "enter contact number" and a "Search" button is located in the top right. Below the header is a large image of a hospital building. The main content area features a sidebar on the left with a menu containing "Appointment", "Payment Status" (highlighted in blue), "Prescription", "Doctors Section", and "Attendance". The main content area on the right contains a form with a yellow input field containing the number "+263776721164", a dropdown menu with "paid" selected, and a blue "update" button.

## 9.5 Result Searching One

The screenshot shows a web browser window with the URL `localhost:8080/hms/search.php`. The page has a blue header with the text "Search Results". Below the header is a table with the following data:

First Name	Last Name	Email	Contact	Doctor	Payment
Tindo	Zhou	tendaizhou@protonmail.com	+263789312078	Doctor2	Pay later

## 9.6 Result Searching Two



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/hms/search.php'. The page has a blue background and is titled 'Search Results'. It contains a table with the following data:

First Name	Last Name	Email	Contact	Doctor	Payment
Tindo	Zhou	tendaizhou@protonmail.com	+263789312078	Doctor2	Pay later

## 10 CONCLUSION

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