SOFTWARE REQUIREMENTS SPECIFICATION

for

Hospital Appointment System

Version 1.0

Prepared by : 1. Chaitanya R.(220970920007)
2. Disha S. (220970920009)
3. K.Nehru(220970920013)
4. Krishna M.(220970920015)
5. Somnath T.(220970920041)

Submitted to : Dr. Ritu Kapur
Mentor

Contents

1 Introduction

In this hectic world, we don't have time to wait in infamously long hospital queues. The problem is, queuing at hospital is often managed manually by administrative staff, we take a token, wait for our turn, then we are directed to meet doctor. What's more exasperate is that we traveled a long distance only to learn the doctor isn't available.

Hospital Appointment System will help us overcome all these problems because now patients can book their appointments at home, they can check whether the doctor they want to meet is available or not. Doctors can also confirm appointments, this appointment system will save time and money of the patient.

The main goal is to automate all the information regarding the patient and the doctors. This will results in better administrative functions and ultimately better patient care, which is the prime focus of any healthcare unit.

1.1 Purpose

This software will help the company to be more efficient in registration of their patients and manage appointments, records of patients. It enables doctors and admin to view and modify appointments schedules if required. The purpose of this project is to make hospital appointment booking managing procedure digitalized.

1.2 Scope

The intention of the system is to increase the number of patients that can be treated . If the hospital appointment system is file based, management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also could be misplaced by losing data and information.

1.3 Perspective

This web application is designed for Hospital Appointment Automation. The patient book their appointment from web application rather than going to the hospital and wasting time in a queue. The software aims to book the appointment of patient through the web page and the patient can check if it is been approved by the Doctor depending upon the availability. The receptionist can also book/delete/update the appointment. The admin has all the rights to add/delete/update Patients/Doctor/Receptionist. The doctor can view/approve the appointments.

Benefits of Hospital Appointment System:

- Provides Patient Registration form.
- Helps patients cut the long queue and saves their time,a. Is equipped with features like automated email
- Patient can Login and can book/view the appointments.
- The Admin can access and update/delete/add doctors, receptionist.
- Provides Graphical User Interface (GUI) For Doctor to look the Patient List and approve.
- Receptionist have Graphical User Interface (GUI) to monitor doctor and patient schedule and manage their appointments.

1.4 Product Function

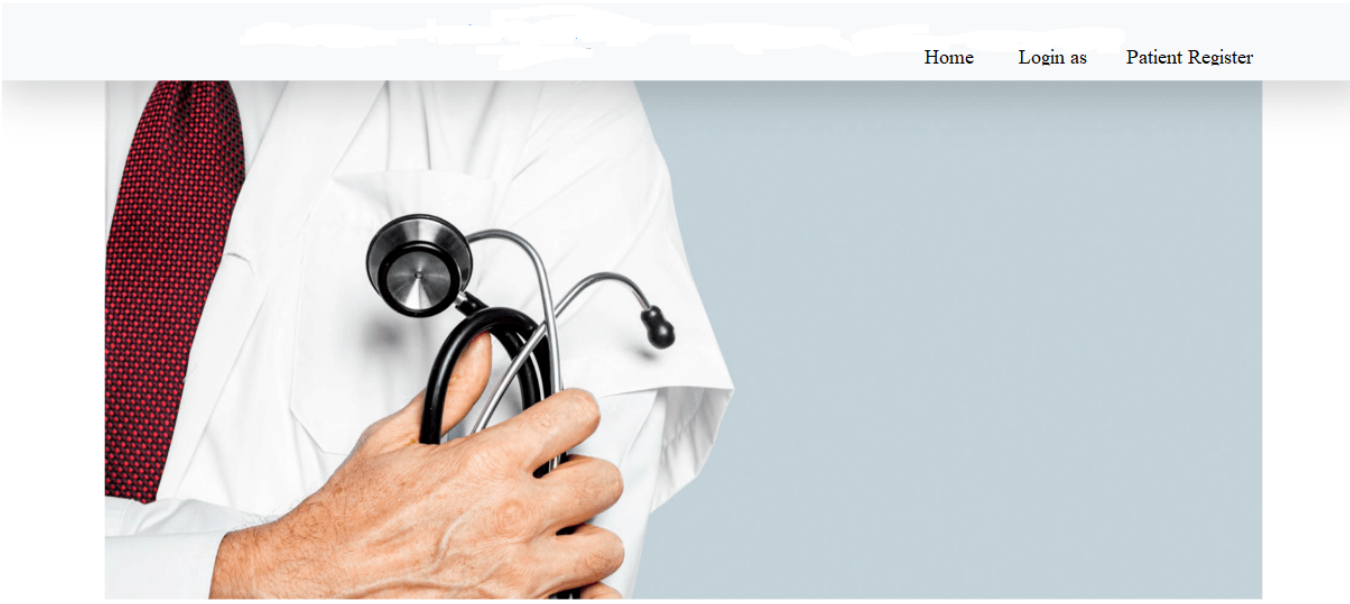

Figure 1.1: Sample Demo

1.5 User Characteristics

1.5.1 Registration:

A new user can register herself/himself into the system and will get a unique ID and password. Like patient can register by filling few personal details like name, address, phone number, email ID.

1.5.2 To Log – In into system:

If the user is registered into the system then he/she can login into the system using their own unique Id and password.

1.5.3 Book Appointment Request:

Patient Login Into the System And Entered Booking Details like name , time, Type of Doctor, Etc... Click On the Submit Button And Send Request To Doctor Approve their Appointment

1.5.4 Helpline:

To connect with Hospital team through Contact service. patients who does not have access to internet service or does not have any knowledge of using web application can also call to Hospital service provider for asking For Booking. Helpline Having Their Hospital Number To Contact Ask For Booking.

1.5.5 Notification:

After submitting the Appointment Booking form, the Patient can view all the notification for his/her own account like Request Send Successful, Request Pending, Request Accepted, Request rejected.

1.5.6 Response from Hospital:

When member of Hospital team login into the system can see all the requests received at their end from Patients. Their task is to accept, process the request and sending acknowledgment to the Patient.

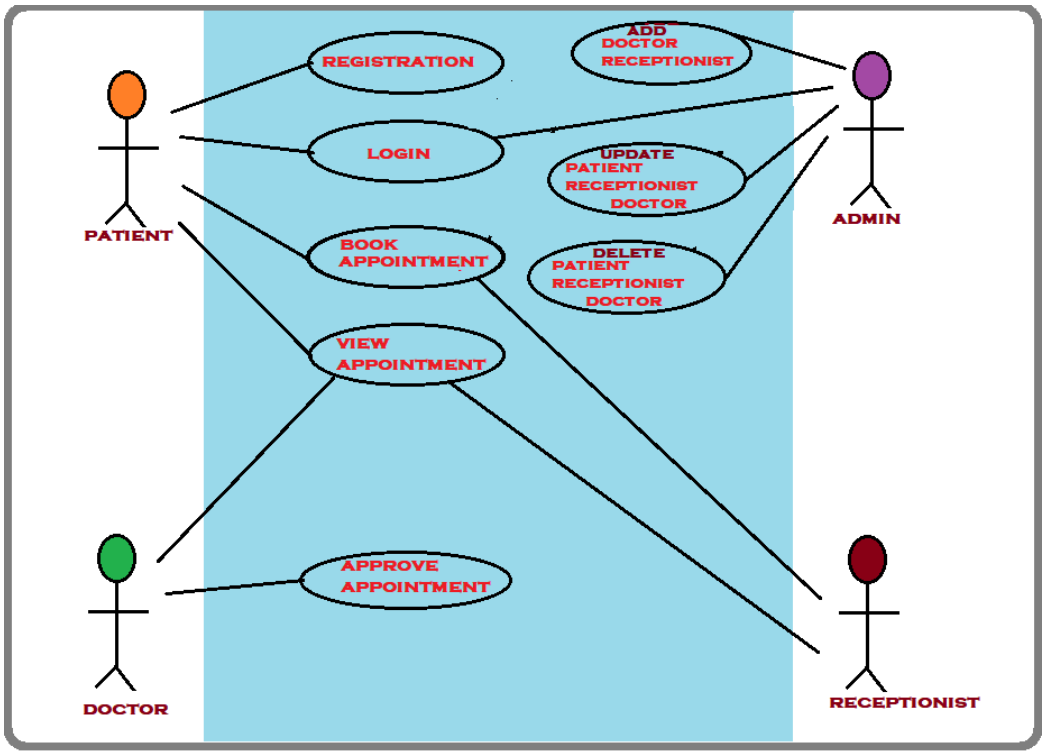

Figure 1.2: UML

2 Requirements

2.1 External interfaces

2.1.1 System interfaces

The application runs in the latest version of Chrome or Firefox browser on Windows, Linux and Mac.

2.1.2 User interfaces

The web page shall provide a very intuitive and simple interface to the patients and other users, so that one can easily navigate through pages, user's registration, login process whereas client can easily manage and revoke other user's permissions.

Graphical Interface

That enables a person to communicate with a computer through the use of symbols, visual metaphors, and pointing devices.

Menu-driven Interface

Our web application provides with a range of options in the form of a list or menu displayed in Fullscreen, pop-up, pull-down, or drop-down.

Form-based Interface

Used to get data from users and save into the database .

2.1.3 Hardware interfaces

Server side

The web application will be hosted on a web server which is listening on the web standard port ****.

Client side

Monitor screen – the website shall display information to the user via the monitor screen Mouse – the website shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus. Keyboard – the website shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

2.1.4 Software interfaces

Server side

An Apache web server will accept all requests from the client and forward it accordingly. A database will be hosted centrally using MySQL.

Client side

An OS which is capable of running a modern web browser which supports JavaScript and HTML5. The Web page is downloaded from the Web server and the user can interact with this content in a Web browser, which acts as a client.

2.1.5 Web- Based Interface

Web browser:

Our website works on most popular web browsers i.e. Google Chrome, Microsoft Edge (formerly Internet Explorer), Mozilla Firefox, and Apple's Safari and works ok with all versions or just on a new one.

Communication standards and Network server communications protocols used:

The HTPP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

Electronic forms:

HTML Forms for user registration and also to get data from patient requesting Appointment booking for different types of Hospital.

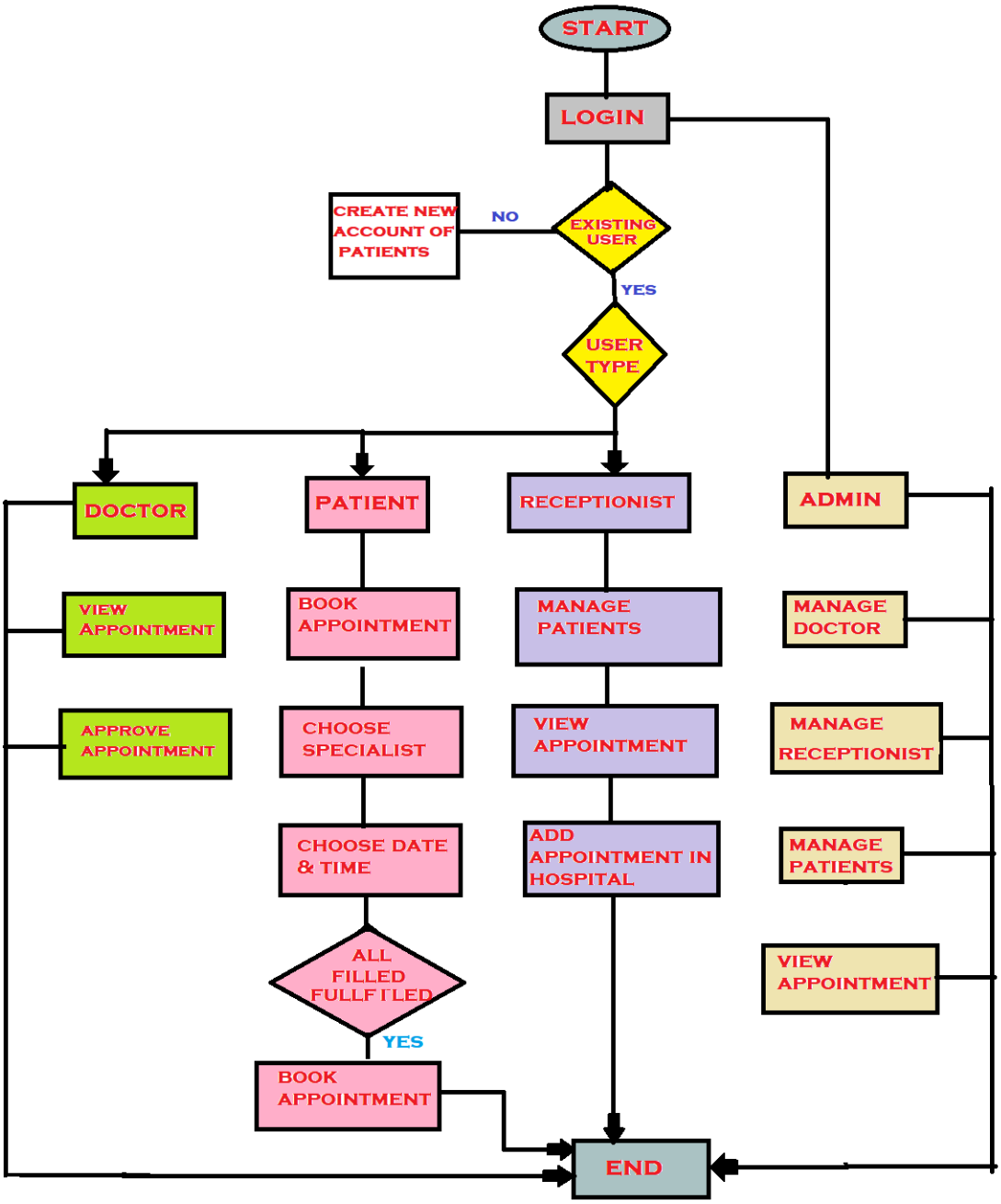

Figure 2.1: Control Flow

2.2 Functions

2.2.1 Functional Requirements:

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

1. **To Log – In into system:**To Log – In into system: Admin, Doctor , Patient, Receptionist
   Input: Click on log-in button.
   Output: User logged in to its account

2. **Sign Up:** Patient create an account
   Input: Patient should Fill up the form with personal details and click on submit button
   Output: A popup will come with confirmation of account created.

3. **24 * 7 HELPLINE service:**
   Patient can connect with hospital team through helpline service at any time.
   Input: Call on the Helpline number.

4. **Notification:**

   To view all the notification for Patient own account like Successful, Request Accepted Request rejected.

2.2.2 Non- Functional Requirements:

These are basically the quality constraints that the system must satisfy according to the project contract. They are also called non-behaviour requirements.

1. Authentication of user whenever he/she logs into the system .
2. The processing of each request send by the Patient should be done.
3. Notification for each request should be sent with a latency of not greater than 12 hours from such an activity.
4. The site should load even when the number of simultaneous users is high

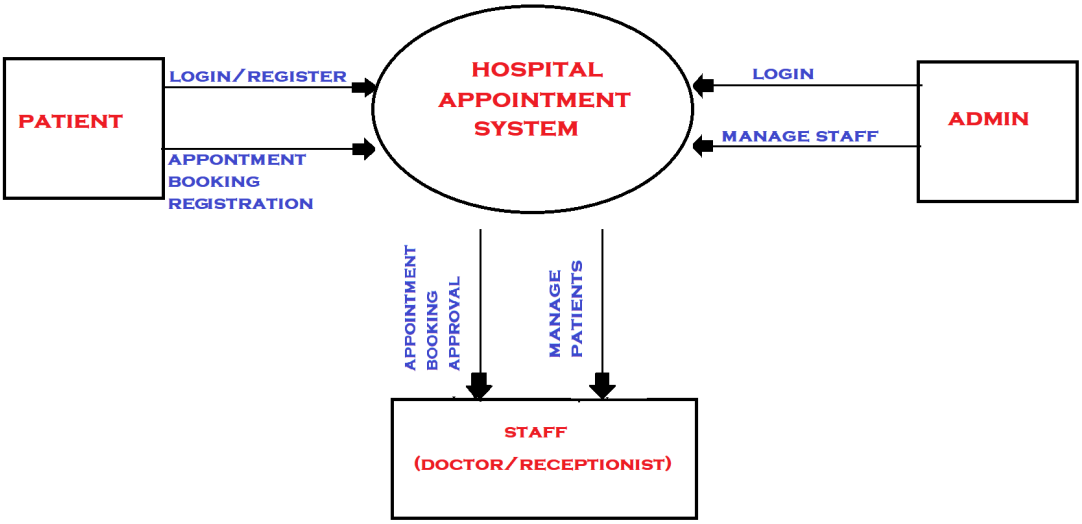

Figure 2.2: 0-level DFD

5. This application should run in any Windows, MacOS and Linux platform having browser of any version. It is also available in mini browsers for mobile

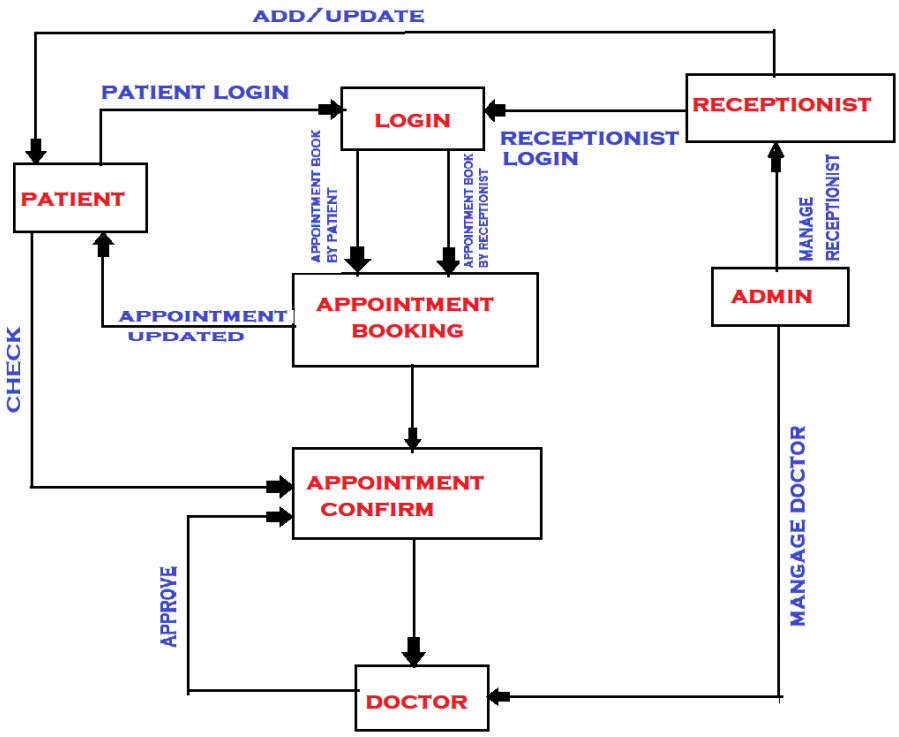

Figure 2.3: 1-level DFD

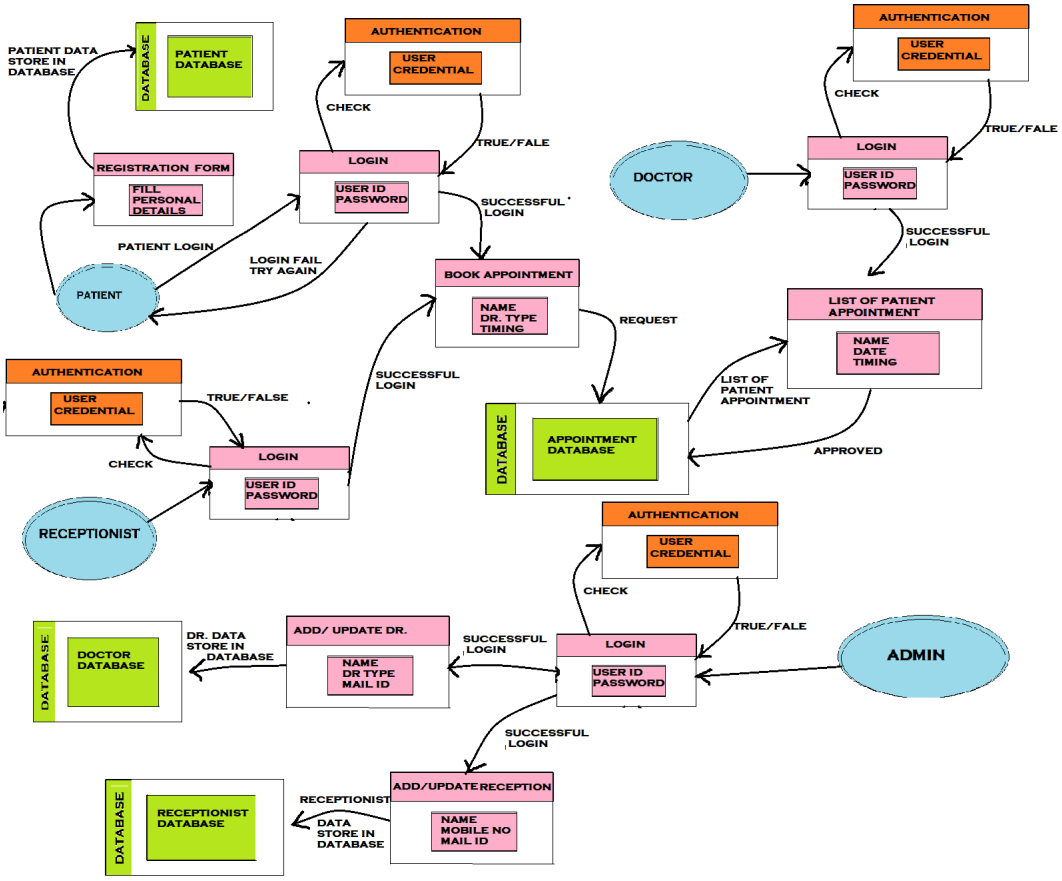

Figure 2.4: 2-level DFD

2.3 Usability requirements

The design should support the following requirements for its primary users:

- **Learnability -** Easy-to-learn interface with simple navigation. All headings, buttons, and error messages are easy to understand.
- The most important thing is to be placed in the right place on a web page like emergency team number, Patient Appointment request, Ambulance service number.
- Give correct choices to the users, in a very obvious way.
- **Navigability** −User will able to perform operations without having to navigate through multiple pages/links. No operation will require more than 5 to 8 clicks.
- **Familiarity** −The system's interfaces and navigations will be based on other systems that the users are familiar with.
- The system will not require any administration tasks at the user level.
- **Help -**The system will be equipped with Computer based tutorial in English and other languages for users to "self-learn" and "self-solve" any navigability or operational doubts.
- **Memorability -**When users return to the website after a period of not using it, he/she can easily re- establish proficiency.

2.4 Performance requirements

- **Start-up Time -** The application should display the opened document within 10s after it is started.
- **Edit Response Time -**The application should display updated values within 1s after user triggers the edit operation.
- **Smooth Scrolling -**While a user scrolls the requirements table, the application should not display scrolling jerks longer than 200ms.
- Since this software is going to web based, it does require a powerful server machine with high band internet access so that it can handle multiple users at the same time.
- The web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections.

- To make the web application lightweight, simple libraries and tools should be used at developing phase.
- System should be able to deal with multiple users at the same time. Also, database of the system should handle at least a thousand of users at any periods.
- To improve portability, software should run on variety of platforms and variety of connection speeds. Portability also means running on different platforms without an additional effort. To achieve this, web application should be developed by using the common technologies and tools which are provided by all common web browsers and operating system.

2.5 Logical database requirements

A Logical Database is a special type of ABAP (Advance Business Application and Programming) that is used to retrieve data from various tables and the data is interrelated to each other. Logical Database we will use joins instead of multiple SELECT statements, which will improve response time and this will increase the Performance of Logical Database.

2.5.1 Below is some important task of Logical Database:

- With the help of the Logical database, we can read the same data from varies programs.
- A database provides the same user interface for multiple programs.
- Database ensures the Authorization checks for the centralized sensitive database.
- When the structure of data is large it is convenient to store it in database.
- We can easily retrieve, modify, save, delete the data using logical database.
- Different functional operation can be performed to retrieve required data from database Like select, join, group by etc.

2.5.2 Designing of Database:

ER model helps to systematically analyse data requirements to produce a well-designed database.**ER Diagram** stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

2.5.3 Components of the ER Diagram

- **Entities:**A real-world thing either living or non-living that is easily recognizable and non-recognizable. An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.
- **Attributes:**It is a single-valued property of either an entity-type or a relationship-type.
- **Relationships:** Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department.
- **Cardinality:**Defines the numerical attributes of the relationship between two entities or entity sets. Different types of cardinal relationships are One-to-One Relationships, One-to-Many Relationships, May to One Relationships Many-to-Many Relationships.

Following are the main components and its symbols in ER Diagrams:
Following are the main components and its symbols in ER Diagrams:

- **Rectangles:**This Entity Relationship Diagram symbol represents entity types.
- **Ellipses :**Symbol represent attributes.
- **Diamonds:** This symbol represents relationship types.
- **Lines :**It links attributes to entity types and entity types with other relationship types.
- **Primary key :**attributes are underlined.
- **Double Ellipses :** Represent multi-valued attributes.

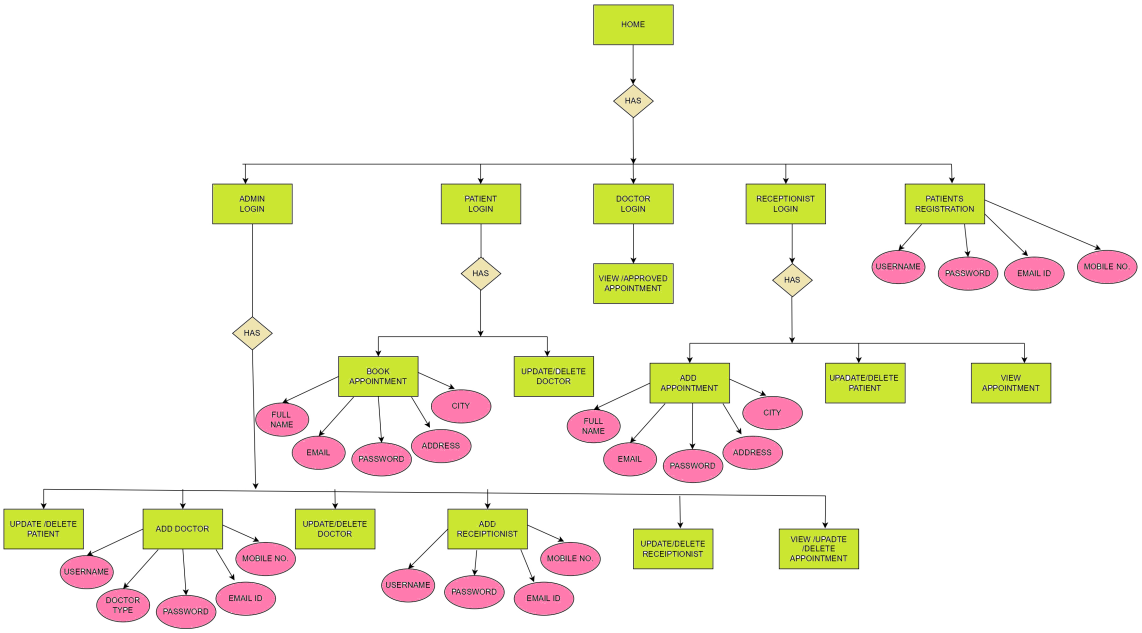

Figure 2.5: ER Diagram

2.6 Design constraints

- **User Interface Constraints:**Using this system is fairly simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.
- **Hardware Constraints:**The system should work on most home desktop and laptop computers which support JavaScript and HTML5.
- **Software Constraints:**: The system will be intended to run on Firefox, Google Chrome and Internet Explorer.
- **Data Management Constraints:**System shall be able to interface with other components according to their specifications.
- **Operational Constraints:**The system is limited by its operating server in terms of the maximum number of users it can support at a given time.
- **Server Side:**: The disk space on Server should always be at least 10GB available.
- **Client Side:**The client's browser must be a modern browser and support HTML5 standards.

2.7 Software system attributes

The software must consist of the following attributes:

- **Reliability**
  The reliability of the overall system depends on the reliability of the separate components. A powerful server should be able to handle multiple requests. The server also should be able to process each request and serve back by sending a response.

- **Availability**
  The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from server and saved by the client.

- **Security**
  Passwords will be saved encrypted in the database in order to ensure the user's privacy and user's IP will be logged. The system will be protected against vulnerabilities like injection attacks

- **Maintainability**
  Database is used for maintaining the user's data records and the server takes care of the site. In case of a failure, a re-initialization of the program is recommended.