

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

## DOCTOR MANAGEMENT SYSTEM

Version 1.0

Prepared by : 1. Kayala Raana Pramodh (200001035)  
2. Koneti Anuhya (200001037)  
3. Kothuru Sharvani (200001038)  
4. Maddaka Yashwanth (200001040)  
5. Nelavalli Sri Nikhitha (200001052)

Submitted to : Dr. Puneet Gupta  
Lecturer

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUCTION</b>                                 | <b>3</b>  |
| 1.1      | PURPOSE . . . . .                                   | 3         |
| 1.2      | DOCUMENT CONVENTIONS . . . . .                      | 3         |
| 1.3      | INTENDED AUDIENCE AND READING SUGGESTIONS . . . . . | 3         |
| 1.4      | PROJECT SCOPE . . . . .                             | 3         |
| 1.5      | REFERENCES . . . . .                                | 3         |
| <b>2</b> | <b>OVERALL DESCRIPTION</b>                          | <b>4</b>  |
| 2.1      | PRODUCT PERSPECTIVE . . . . .                       | 4         |
| 2.1.1    | Appointment Details . . . . .                       | 4         |
| 2.1.2    | Patient description . . . . .                       | 4         |
| 2.1.3    | Appointment Booking description . . . . .           | 4         |
| 2.2      | PRODUCT FEATURES . . . . .                          | 5         |
| 2.3      | USER CLASS and CHARACTERISTICS . . . . .            | 6         |
| 2.4      | OPERATING ENVIRONMENT . . . . .                     | 7         |
| 2.5      | DESIGN and IMPLEMENTATION CONSTRAINTS . . . . .     | 7         |
| 2.6      | ASSUMPTION DEPENDENCIES . . . . .                   | 7         |
| <b>3</b> | <b>SYSTEM FEATURES</b>                              | <b>8</b>  |
| 3.1      | DESCRIPTION and PRIORITY . . . . .                  | 8         |
| 3.2      | STIMULUS/RESPONSE SEQUENCES . . . . .               | 8         |
| 3.3      | FUNCTIONAL REQUIREMENTS . . . . .                   | 8         |
| <b>4</b> | <b>EXTERNAL INTERFACE REQUIREMENTS</b>              | <b>10</b> |
| 4.1      | USER INTERFACES . . . . .                           | 10        |
| 4.2      | HARDWARE INTERFACES . . . . .                       | 10        |
| 4.3      | SOFTWARE INTERFACES . . . . .                       | 10        |
| 4.4      | COMMUNICATION INTERFACES . . . . .                  | 10        |
| <b>5</b> | <b>NON FUNCTIONAL REQUIREMENTS</b>                  | <b>11</b> |
| 5.1      | PERFORMANCE REQUIREMENTS . . . . .                  | 11        |
| 5.2      | SAFETY REQUIREMENTS . . . . .                       | 12        |
| 5.3      | SECURITY REQUIREMENTS . . . . .                     | 12        |
| 5.4      | RESPONSE TIME . . . . .                             | 12        |
| 5.5      | STORAGE SPACE . . . . .                             | 12        |
| 5.6      | SOFTWARE QUALITY ATTRIBUTES . . . . .               | 13        |

# 1 INTRODUCTION

## 1.1 PURPOSE

The Purpose of this document is to build an online system to manage doctors and their patients and their appointments

## 1.2 DOCUMENT CONVENTIONS

This document uses the following conventions.

|     |                      |
|-----|----------------------|
| DB  | Database             |
| DDB | Distributed Database |
| ER  | Entity Relationship  |

## 1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project serves as a proof of concept for a doctor management system. This website can be used to consult a doctor by anyone. Both doctors and patients will benefit from this endeavor.

## 1.4 PROJECT SCOPE

The goal of the online doctor management system is to make it easier for doctors to run their practices while also providing a convenient and user-friendly application for patients looking for an appointment. With its doctor management and reservation functionalities, the system is built on a relational database. We'll have a database server that can support hundreds of major cities throughout the world, as well as thousands of doctors from different hospitals. Above all, we strive to deliver a pleasant user experience and solicit patient feedback.

## 1.5 REFERENCES

<https://krazytech.com/projects>

## **2 OVERALL DESCRIPTION**

### **2.1 PRODUCT PERSPECTIVE**

A doctor management database system stores the following information.

#### **2.1.1 Appointment Details**

It provides the appointment's starting and ending timings, as well as the number of other patients that are scheduled in the same time slot as you.

#### **2.1.2 Patient description**

It includes Patient's name,address,blood group,past medical records. This information may be used for keeping the records of the patient for any emergency or for any other kind of information.

#### **2.1.3 Appointment Booking description**

It includes patient details,appointment slot details,consulted doctor details, date of booking, date of appointment.

## 2.2 PRODUCT FEATURES

The major features of doctor management database system as shown in below entity–relationship model (ER model)

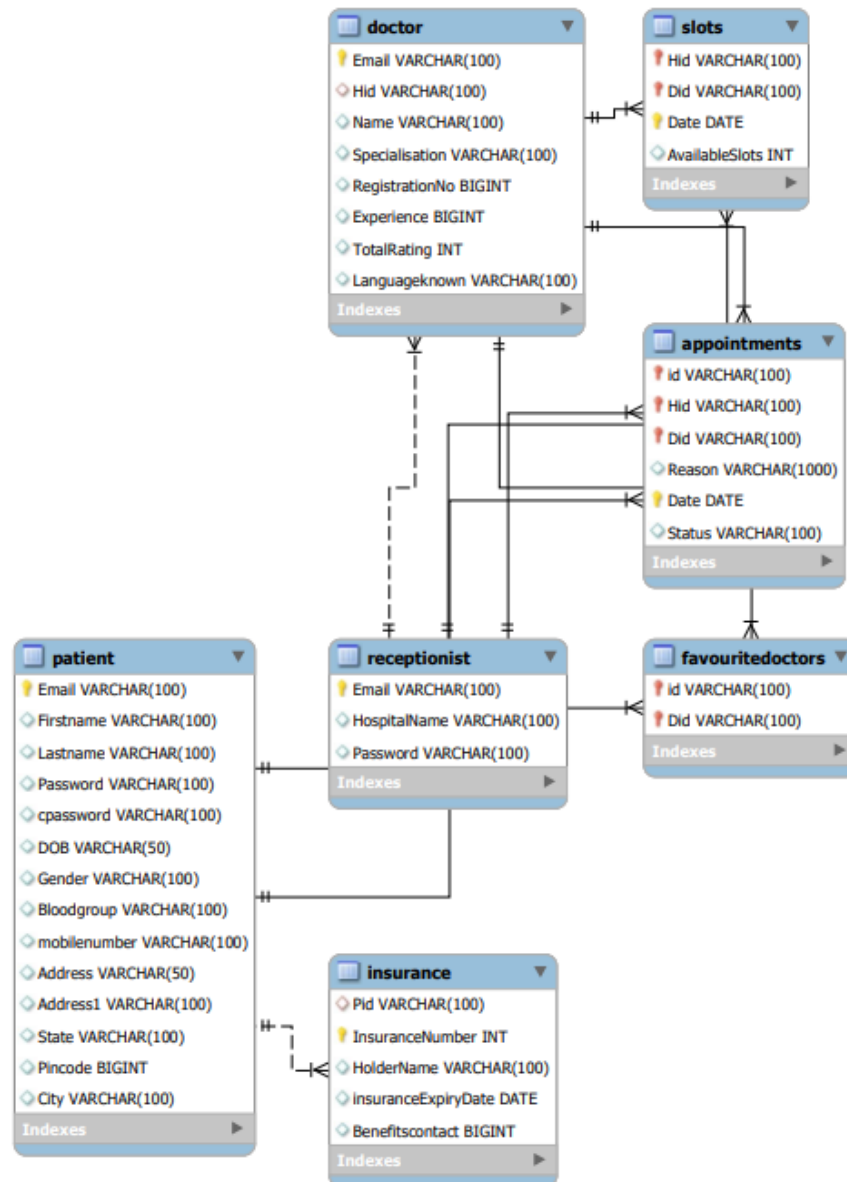


Figure 2.1: The diagram shows the layout of doctor management database system – entity–relationship model

## 2.3 USER CLASS and CHARACTERISTICS

Users of the system should be able to get from the database information like specific doctor's slot, specific doctor's qualification and practise, and patient's appointment details. Patient, Receptionist, and Employee user privileges will be supported by the system. Patients will be able to access Patient functions, Receptionists will be able to access Receptionist functions, and Employees will be able to access all patient, receptionist, and appointment management features.

### **The patient should be able to do the following tasks:**

- Make a new appointment
- Cancelling an existing appointment
- View his appointment status(whether consulted doctor is on leave on that appointment slot or not)
- Check doctor's profile (Users can choose doctors from the shown list by ratings also.)
- Provide insurance details
- Update his Medical records
- Create a preference list where patient can choose his favourite doctors

### **The receptionist should be able to do the following tasks:**

- Mark no show appointments
- Provide details of doctors
- Provide details of slots

### **The following management functions should be available to the employee:**

#### **Customer Functions**

- Get all patients who have booked appointment slots.
- Get list of doctors of a hospitals.
- View appointment schedule.
- Get all doctors who are on sudden leave.

#### **Administrative**

- Add/Delete a doctor
- Add a new hospital
- Replace the doctor who is on unexpected leave with a fresh doctor.
- Add a new appointment time that is convenient for the specific doctor who is on unexpected leave.
- Update appointment times based on replacement slots (which arise as a result of a specified doctor's unexpected absence).

A doctor can see a maximum of 5 patients in a given time period.

## 2.4 OPERATING ENVIRONMENT

Operating environment for the DOCTOR MANAGEMENT SYSTEM is as listed below.(Include the details as per your application)

- Operating system: Windows
- database: MYSQL database
- platform: VScode

## 2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

- We have used MYSQL server.
- The information of all patients, doctors, appointments will be stored in a database.
- It is a web based application.
- SQL queries are used to retrieve the data from database.
- Users and Receptionist must login with their correct username and password to login into their accounts.
- This website runs 24hrs a day.
- Users can access the website from any computer with active internet connection.

## 2.6 ASSUMPTION DEPENDENCIES

Let us assume that doctor management system is used in the following application:

- A request for booking/cancellation of an appointment,in case,there are no available slots at specific time ,then user will be provided details of other available slots on same day, or available slots of same doctor in other days
- Based on the patient reviews,calculating appropriate rating points for doctors.
- A doctor can see a maximum of 5 patients in a given time period.

## 3 SYSTEM FEATURES

### 3.1 DESCRIPTION and PRIORITY

The Doctor Management system maintains information about Doctors, Patients, Slot availability etc. One can book/cancel their appointments easily from any place using this project.

### 3.2 STIMULUS/RESPONSE SEQUENCES

- Search for Doctors with their specialization or Hospital name.
- Displays a detailed list of available slots for a week.
- Users can easily reserve the slot at a particular time.
- Cancel an appointment if required

### 3.3 FUNCTIONAL REQUIREMENTS

#### REGISTER/LOGIN

We will be providing two portals. One for Receptionists and the other for the users. Users have to register themselves by providing their personal information like name, age, Gender, Blood group, and contact details.

For the Receptionists, we will be asking the Hospital details and the details of doctors working at their hospitals.

We will ask for a unique UserID and password for each user to log in to the account. Similarly, for each Hospital there will be a separate Receptionist ID and password.

#### AFTER USER LOGIN

Once the user logs into his account, he/she can find various options like search, appointments, Insurance, Favourites, and Records.

**Search Option-** Users can search for the doctors by name, specialization, and Hospital name. Based on the search, a list of doctors will be shown on the screen. Users can choose doctors from the shown list by ratings also. When a user chooses a doctor from the list, the Profile of that doctor will be displayed with his details and feedback given by his previous patients. If the user wishes to book an appointment with that doctor, there will be a “Book Appointment” option. On clicking the option, the user will be directed



to a page where the availability of slots for next week will be shown. He can book the slot by clicking on the date and providing details like a reason for visit(optional), past medical reports(if available).

**Appointments-** In this option, the user can see his future appointments. He can also cancel his appointments from here. We will be providing a “Cancel Appointment” option to cancel the appointments.

**Insurance-** The user can provide his medical insurance details here.

**Favourites-** If a user marks his preferred doctors, they will appear here and he can directly select them from here. Frequently contacted doctors will also be shown here.

**Records option-** If a user wishes to save his reports for the future, we can provide him with a Records option where he can upload and save them for later use. Once the consultation is done, we will also provide a Feedback option for the user to give Feedback to the respective doctor.

## **AFTER RECEPTIONIST LOGIN**

Once the Receptionist logs into the account, there will be options like Add doctors, Appointments.

**Add doctors Option-** To add doctors, click on add option where they will be directed to a page where all the details of the doctor have to be submitted.

**Appointments Option-** In this section, They can see all the appointments linked with their hospital and they can make changes regarding the slot availability of the doctor. When they click on an appointment, the profile of the patient will appear to them. They can also mark No show appointments from here. If a doctor keeps leave due to some work, then the user who booked an appointment for that doctor will be given the option either to cancel or choose a different date from the available slots or choose a different doctor on the same day.

## 4 EXTERNAL INTERFACE REQUIREMENTS

### 4.1 USER INTERFACES

- Front-end software: VS code
- Back-end software: MYSQL

### 4.2 HARDWARE INTERFACES

- Windows

### 4.3 SOFTWARE INTERFACES

The software used for the doctor management online application is listed below. (Include the software specifications relevant to our project.)

| Software used    | Description  |
|------------------|--|
| Operating system | We selected the Windows operating system because it provides the maximum support and it is the most user-friendly. |
| Database         | To save the appointment records, patients records we have chosen MYSQL database.                                   |
| VScode           | We selected VScode for its more interactive help to implement the project.   |

### 4.4 COMMUNICATION INTERFACES

Our project supports all types of web browsers. We are using simple electronic forms for the registration forms,login forms appointment booking etc.

# 5 NON FUNCTIONAL REQUIREMENTS

## 5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of appointment database are as listed below

### E-R DIAGRAM

The E-R Diagram is a way for visually expressing the logical structure of a database. This analysis is then used to arrange data into a relation, normalise the relation, and create a relation database.

**ENTITIES:** These are the different real-world things that are specified in an application.

**PROPERTIES/ATTRIBUTES:** These define an entity's and relationships' properties and attributes.

**RELATIONSHIPS:** These are the connections that connect entities and represent meaningful relationships between them.

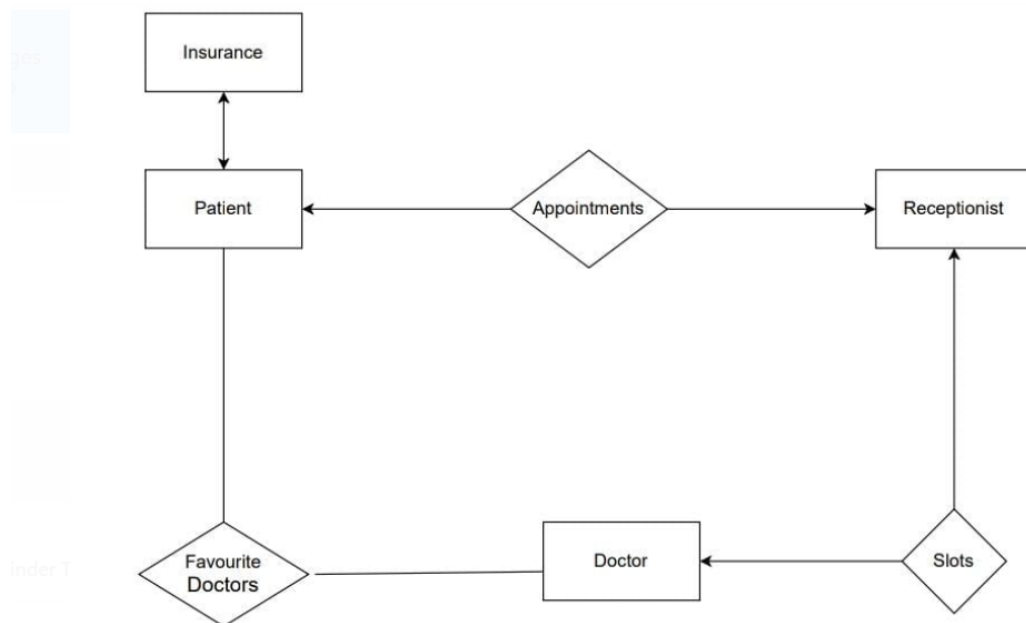


Figure 5.1: The diagram shows the **ER diagram** of doctor management database

## **NORMALIZATION**

The primary goal of normalisation is to reduce redundancy, which implies that information should only be saved once. Storing information many times wastes storage space and increases the total amount of the data saved.

Modification anomalies can occur if a database is not correctly constructed. When data is added to, updated in, or deleted from a database table, modification anomalies occur. Similarly, data redundancy may be an issue in both traditional databases and incorrectly built relational databases. Normalizing a database can help to eliminate them.

The process of dividing a table into smaller tables is known as normalisation. As a result, each table focuses on a specific subject. There are three types of anomaly modifications, and the first, second, and third normal forms (3NF) are regarded acceptable for most practical purposes. It should be considered only after a thorough examination and comprehension of its implications.

## **5.2 SAFETY REQUIREMENTS**

In case of any failure which causes severe damage to the database, the previous copy of the database can be restored using recovery techniques. The web framework used in this project is Flask which makes the retrieval of information quick which not only saves time but also helps in the proper functioning of the software.

## **5.3 SECURITY REQUIREMENTS**

Flask allows us to add many security features such as Session-based authentication, Role management, and Password Hashing.

## **5.4 RESPONSE TIME**

It's a measure for how quickly a server replies to a user's request to open a website under normal conditions (number of website visitors are fixed) for suppose if there is an increase in the number of website visitors then flask adjust the visitor's range and responds accordingly.

## **5.5 STORAGE SPACE**

This prevents stack overflow. This helps in minimizing the response time, fastening the processing, determining the behavior of a system when some existing device is upgraded in the software, and also identifying the practical storage limit before getting deployed.

## **5.6 SOFTWARE QUALITY ATTRIBUTES**

### **MAINTAINABILITY**

Flask provides a large number of extensions that make it simple to add new functionalities in order to meet users' requirements and it will be easier to maintain. The administrators and the receptionists should maintain the updated appointment schedule so that new appointments can be done accordingly.

### **PORTABILITY**

As Flask is portable this software can be used in various environments. User can book appointments in their preferred platforms.

### **FILEXIBILITY**

Flask has an extensible web framework. It gives us the flexibility to customize our applications the way we want. This allows us to reduce issues that could arise due to the rigidity of other frameworks. If there are any latest features added to the website then user has ability to access it.

### **SCALABILITY**

It can be achieved by creating an efficient database so that the system does not get overloaded even if there are so many users using the website at the same time. In case if there are many website visitors at once then website doesn't get overloaded.

### **UTILITY**

Flask is a toolkit that can implement requests, analyze, configure, monitor, which is an easier way to software utility.

### **CAPACITY**

This website supports more users to use the website at once. Once the patient's information is validated, the website gives confirmation in a matter of seconds. If patient wants to access his/her medical records this can be done easily.

### **AVAILABILITY**

It is a property of software that it is always present and ready to do its function when user needs it. The appointments should be available on the specified date and specified time as many patients are doing advance reservations.

## **USUABILITY**

The degree to which a system makes it simple for a user to achieve that purpose. The appointment schedules should satisfy a maximum number of patients needs.