**DOCTOR PATIENT MANAGEMENT SYSTEM**

High Level Design

**Document Control :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Doctor Patient Management System** | | | | | | | | |
| Guided by- **Mr. Biswa P Das** |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| October 20,2022 | 1.0 | Hemanth Ketha,  K.Maneesha  Kanika Saini  E.Noshnavi  P.Sushma |  | | | |  | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |

# Introduction

The aim of this document is to gather, analyze and give an in-depth insight into the complete Doctor Patient Management System by defining the problem statement in detail. The detailed requirements of Doctor Patient Management System application is provided in this document.

**1.1 Intended Audience:**

This document is intended to be read by Client.

**1.2 Acronyms/Abbreviations:**

|  |  |
| --- | --- |
| CLIENT | USER |
|  |  |
|  |  |

**1.3 Project Purpose:**

The purpose of this document is to describe the requirements for the Doctor Patient Management System. This project helps in easy access to maintaining files and to avail better services to doctor and patients from anywhere.

**1.4 Key Project Objectives:**

1. Create and register for new user

2. Validate credentials

3. Book appointments

4. View appointments

5. File maintenance

6. Update personal information of user

7. View patient records

7. Query doctor appointments list

**1.5 Project Scope**

This project aims to create and develop the doctor Patient Management System which takes patient Aadhar number, personal details and doctor MCI license number, personal details and preference for registration and login. It allows the user to add the personal data and choose the operation to do, for doctor it is to provide slots for an appointment. Similarly for patient, it is to book the appointment with doctor .All the information about the user is stored in a retrievable manner.

**1.6 Functional Overview**

**Doctor\_Registration** : All the details of doctor will be registered and duplicate data will be checked.

**Doctor\_Slot\_Selection:** It will choose the slot which doctor want to give for allotment to the patients.

**Patient\_Registration** : All the details of patient will be registered and duplicate data will be checked.

**Book\_Appointment** : It contains the list of Specialization and slot for the patient to book appointment.

**View\_Bookings** : The doctor can view his appointments and other details when booked by patients. The patient can see his booked appointments with doctor ,like Doctor name, MCI license Number etc.

**Record\_Patient\_History** : It will record the Patient details including treatment, disease, medication etc to the doctor.

**View\_Patient\_History** : It will display the history of patient with details like name, medication etc to doctor

**Doctor\_File\_Maintenance** : The admin can add, delete, view and modify doctor details with this function

**Edit\_Doctor\_Info**:It will allow doctor to edit details like Mobile no., address, slot, e-mail.

**Patient\_File\_Maintenance :** The admin can add, delete, view and modify patient details with this function

**Doctor\_License\_Query** : Admin can view any doctors appointments list with doctors MCI license

**2.Design Overview:**

**2.5 Performance:**

The system will work on the customer terminal. The performance depends on the hardware component of the user’s system.

**2.6 Maintenance:**

Very little maintenance could be required for this setup. An initial configuration will be the only system required interaction after the system is put together.

The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed.

Physical maintenance on the system’s parts may be required, and would result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project but may result in downtime.

**3.Environment Description:**

**3.1 Time Zone Support:** IST- Kolkata

**3.2 Language Support:**  English

**3.3 User Desktop Requirements:**

a.   64-bit processor, 1 GHz or faster

     b.   At least 2 GB free hard drive space

     c.   At least 1 GB RAM

**3.4 Server-Side Requirements:**

      a. 64-bit processor, 1 GHz or faster

      b. At least 1 GB free hard drive space

     c. At least 1GB RAM

**3.4.1 Deployment Considerations:**

a. Easy setup: no session storage daemon, use tmpfs and memory caching to enhance performance.

  b. Local storage is used.

  c. No network latency to consider.

  d. To scale buys a bigger CPU, more memory, larger hard drive, or additional hardware.

**3.4.2 Application Server Disk Space:**

  a. No such disk space is required as the program is fully functional on online  IDE(s) as well. The Local Operating System is required and one text file to store the records of processes.

**3.4.3 Database Server Disk Space:**

C

**3.4.4 Integration Requirements:**

   1. Language:  Cpp

   2. Tools:  Cpp check, Valgrind, Makefile

   3. Complier:  g++

   4. Linux Environment

**3.4.5 Jobs:**

We can establish connections between clients who are connected to the server. And we can search the chat history of the clients.

**3.4.6 Network:** End to End

**3.5 Configuration:**

**3.5.1: Operating System**: Linux environment