**EFFICIENT DOCTOR PATIENT MANAGEMENT PORTAL**

**High Level Design & Low-Level Design**

The purpose of this document is to provide a template for documenting both HLD & LLD.

**Design**

**Version - 0.1**

                                                **Document Control :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | |
|  |  |  |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | **Approver Signature** | |
|  | V0.1 | Group 4 | Initial draft | | |  | |

**TEAM MEMBERS**

|  |  |
| --- | --- |
| **EMP Number:** | **Name** |
| 46289123 | NAVYA SHREE |
| 46289124 | HARSHITHA N |
| 46289125 | GAVIREDDY SIREESHA |
| 46289137 | GOPIKA P |
| 46289126 | PANDETI LASYA |

|  |  |  |  |
| --- | --- | --- | --- |
| **1. INTRODUCTION………………………………………………...5**  **1.1. Purpose………………….…………………………………………….………...…..…….5**  **1.2. Scope………………………...…………………...………………….……..………5**  **1.3. Definition…………………………………………………….…….…..…….5**  **1.4. Overview…………………………………………………………………...………5**  **2. GENERAL DESCRIPTION………………...……………………….5**  **2.1. Product Perspective……………………………………….…………………...…5**  **2.2. Tools used……………………………………...…………….………………6**  **2.3. General Constraints……………………………………...………...……….6**  **2.4. Assumptions………………………………………………...…………….....6**  **2.5. Special Design aspects………………………………………………...…….6**  **3. DESIGN DETAILS………………………………….………...….…6**  **3.1.  Main Design Features…………………………….………………...……...6**  **3.2.  Application Architecture……………………………………………………..……...6**  **3.3.  Standards………………………………………………………………………………….……...7**  **3.4.  Data Flow Diagram…………………………………………………….…………...…...7**  **3.5.  Files………………………………………………………………...……………….….…………...11**  **3.6.  User Interface…………………………………………...………………………...…...11**  **3.7.  Reports……………………………………………………...……………….…………...11**  **3.8.  Error Handling…………………………………………...…………………....………11**  **3.9. Interfaces……………...…………………………………...………………………….11**  **3.10. Help………………………………………………...………………......….11**  **3.11. Performance………………………………………...……………………….11**  **3.12. Security…………………………………………………….…………………………...11**  **3.13. Reliability…………………………………………...………………………………….11**  **3.14. Maintainability…………………………………………...…………………………...12**  **3.15. Portability…………………………………………………………...………………….12**  **3.16. Reusability…………………………………………………………………...…………...……...12**  **3.17. Application compatibility…………………………...…………………...…………...…...12**  **3.18. Resource utilization…………………………………………………...…………………….12**  **3.19. Major Classes………………………………………………...……………….………13**  **4. DETAILED SYSTEM DESIGN…………………………...…...…14**  **4.1.  Design Descriptions……………………………………………………...……………14**  **4.4.  Design and Implementation Constraints………………….……………….…...14**  **4.5.  User Interface…………………………...……………………..……………...….……14**  **5. DEMO………………………………………….…………..........…15** |  |  |  |

**1.Introduction**

**1.1 Purpose**

The Doctor Appointment Management System is based on the concept of making patient appointments easier. The system helps to reduce the problems that occur when using a manual system and helps patients to skip endless queries. Some hospitals provide the opportunity to make appointments by placing the call, but sometimes these calls are left unattended. The proposed system will overcome all these drawbacks of the existing system. The proposed system has many advantages in that it stores all the information regarding patients' details, patient profiles, prescriptions, etc .

**1.2 Scope**

This project provides a scope of comprehensive overview of the Doctor Appointment Management System. It highlights the high level flow / use cases in appointment management system and serves as an input to the low level design documents that would further elaborate on the proposed system design

**1.3 Definitions**

        N.A

**1.4 Overview**

The project in whole gives a brief overview of the Appointment management , where the patient and the doctors have to get themselves logged in into the system. where the patient having a specific disease gets the appointment fixed with the doctor having qualification on that disease .

**2. General Description**

**2.1 Product Perspective**

Aim of this project is to create a doctor patient handling management portal that will help doctors in their work and will also help patients to book doctor appointments and view medical progress. The portal allows doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name. The portal manages the appointment data for multiple doctors of various dates and times. Each time a user visits a doctor his/her medical entry is stored in the database.

**2.2** **Tools used**

1 Platform . – Unix/Linux file portal.

2.Programmin Language - C language

3.Pictorial representations – Canva

**2.3 General Constraints**

Doctor Patient Management Portal is user-friendly and automated .

Which further allows a hassle free appointment between patients and doctors .

**2.4 Assumptions**

 To be discussed.

**2.5 Special Design aspects**

The Design aspects of the portal is that it will work with a single user at a time .

**3.  Design Details**

**3.1 Main Design Features**

The main design features include four major parts:

- The architecture,

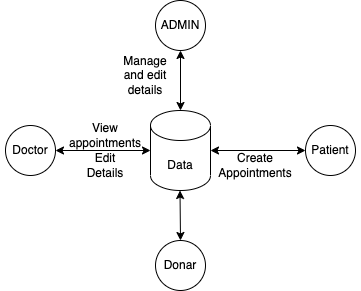
- The user interface design,

- The files,

- Process relation,

- Automation .

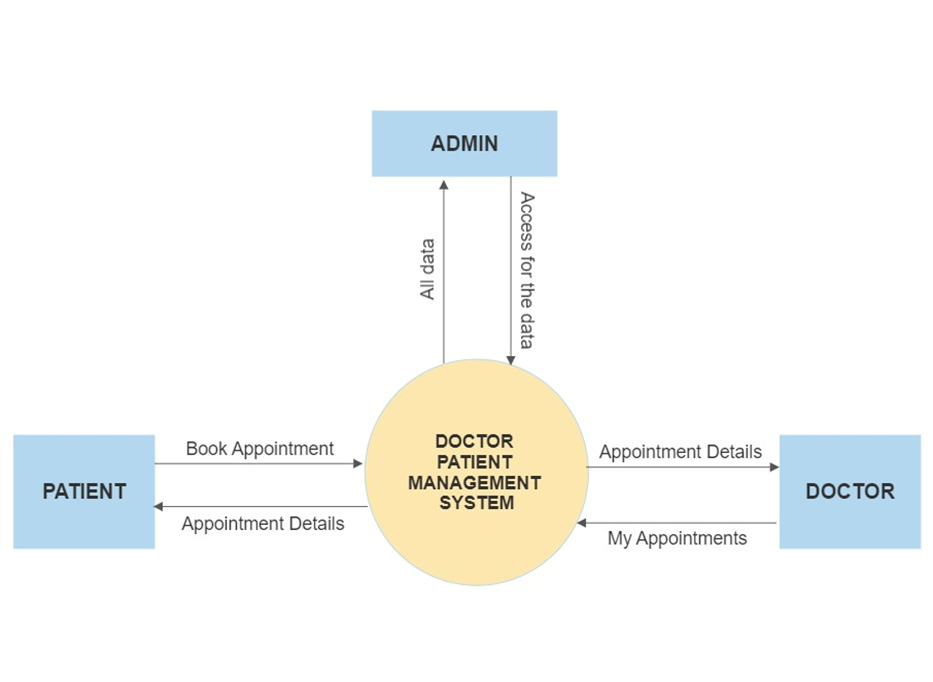
In order to make these designs easier to understand, the design has been illustrated in attached diagrams ( Use Case, Data flow diagrams).

**3.2 Application Architecture**   
  
  
  
  


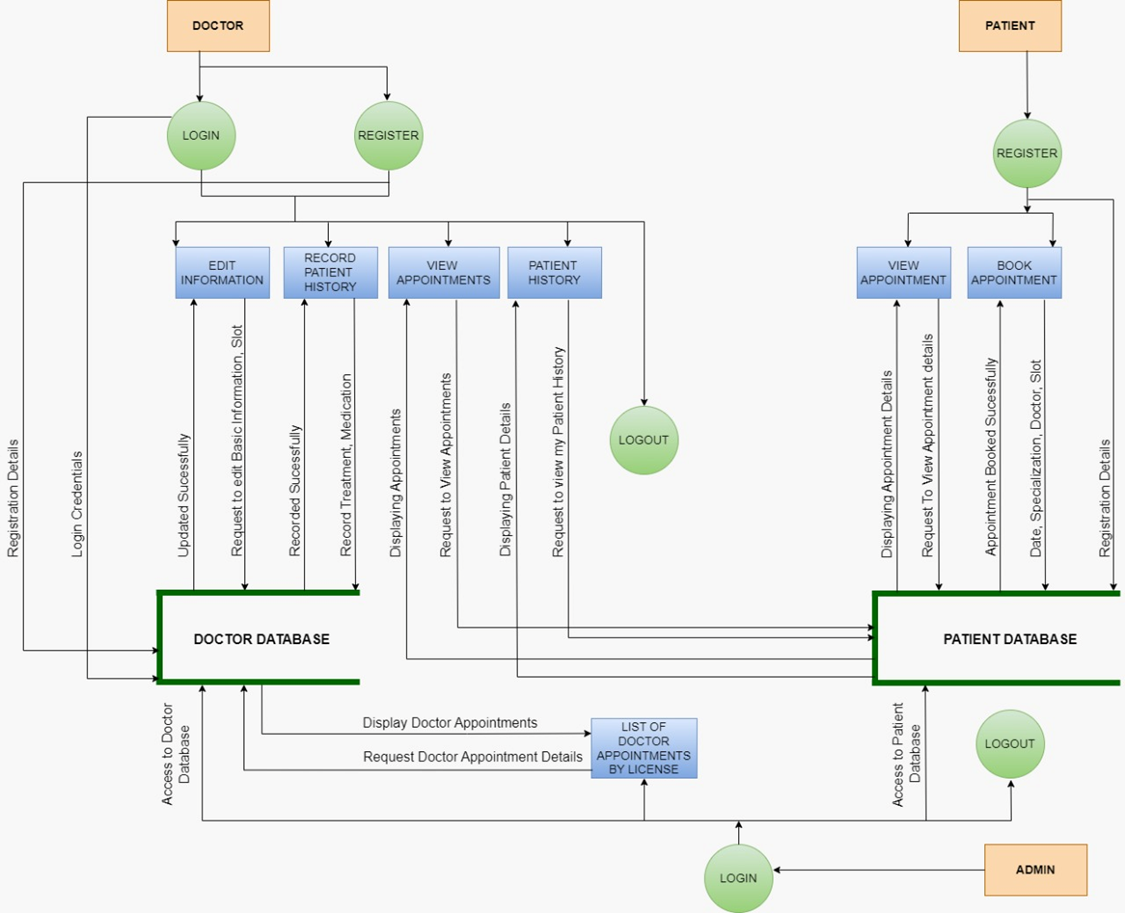
**3.3 Standards**

Security – username and password are required for access to the portal. Quality – by keeping the interface simple and direct, quality should be kept at a maximum.

**3.4 Data Flow Diagram**

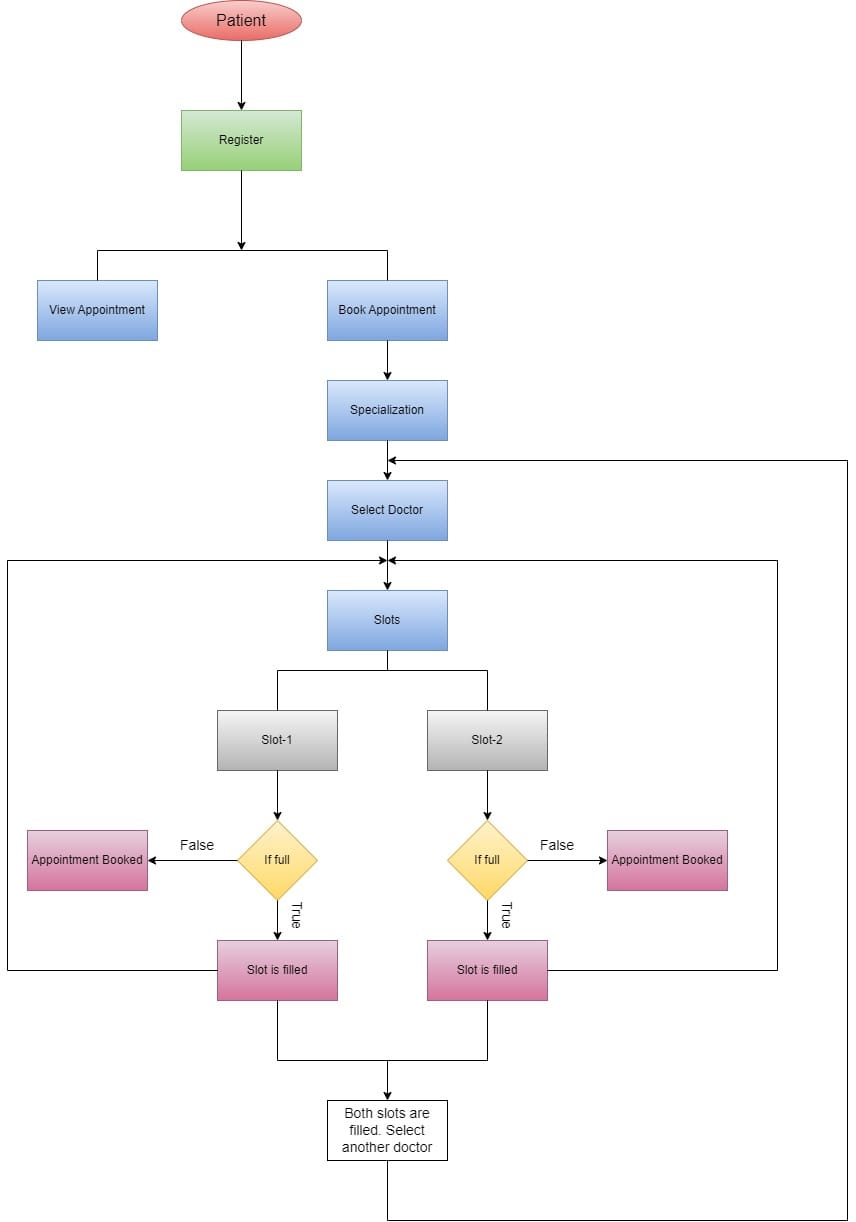
****

**DFD LEVEL 0 DIAGRAM**

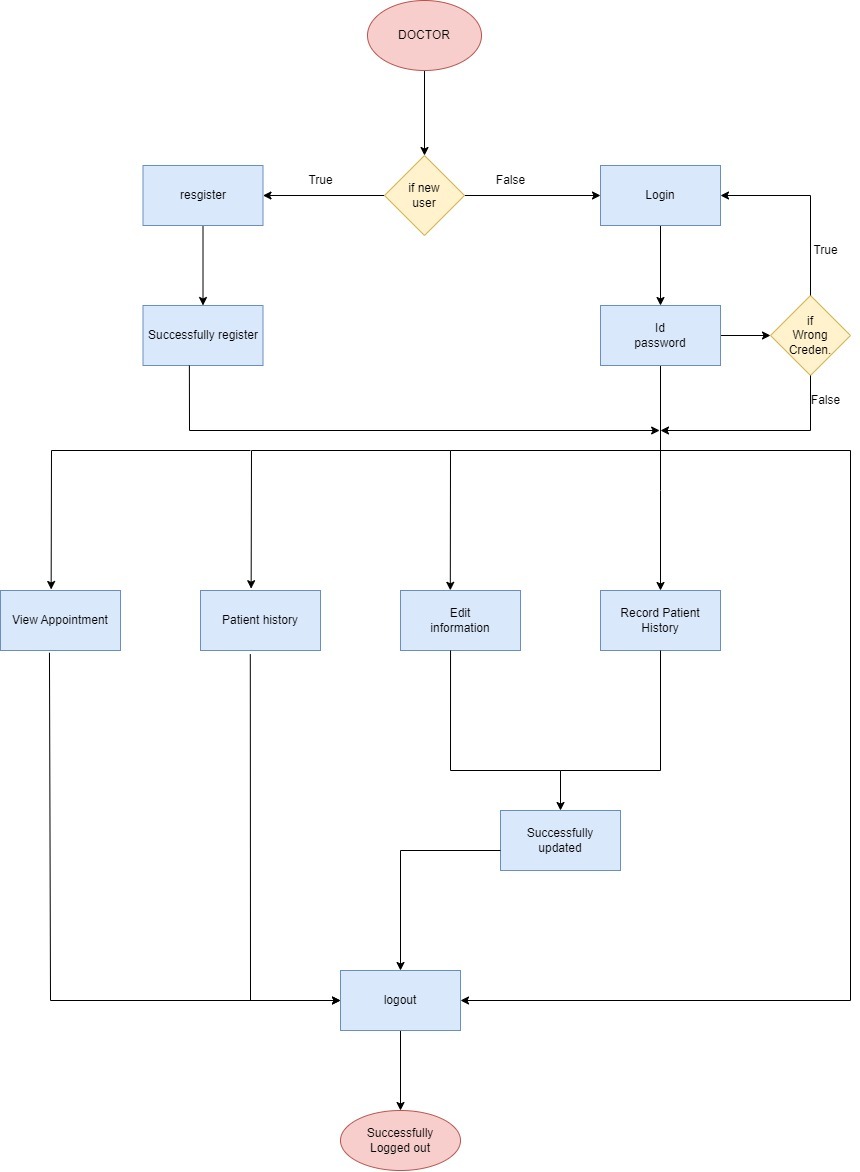
****

**DFD LEVEL 1 DIAGRAM**

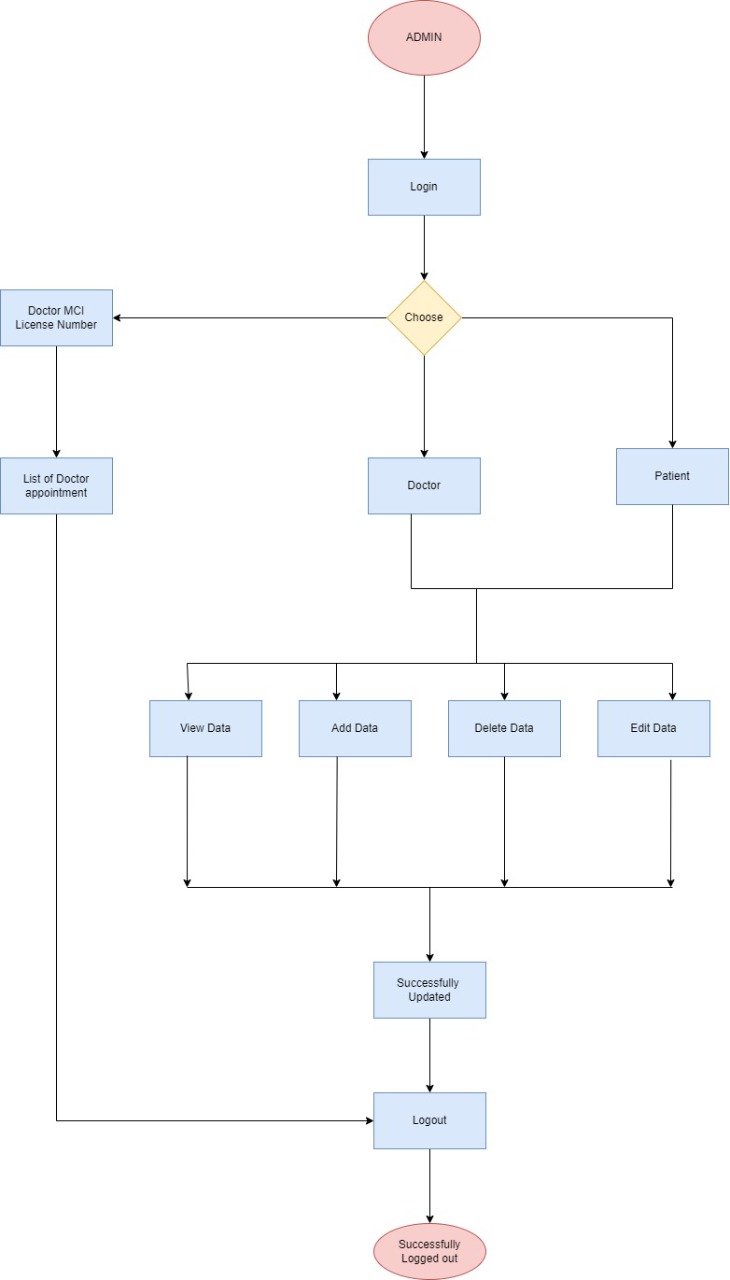
**Flow chart**



**Patient Flow Chart**



**Doctor Flow Chart**



**Admin Flow Chart**

**3.5 Files**

The Doctor Patient Management Portal will use quite a number of files for saving data. It will store login           data, user data, user medical data, prescription, schedule, and appointment details.

**3.6 User Interface**

  File portal interface

**3.7 Reports**

N.A

**3.8 Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong.

An error will be defined as anything that falls outside the normal and intended usage.

**3.9** **Interfaces**

          N.A

**3.10 Help**

Help will come in the form of all the documentation created prior to coding, which explains the intended uses. Should time allow, detailed instructions will be written on how to create and implement the portal with the intention of publishing as an Open Source solution.

**3.11** **Performance**

Performance is going to be very important for this project. For everything to run smoothly for this project, The portal will work on the customer's terminal and the performance depends upon the hardware component of the customer’s portal.

**3.12 Reliability**

The portal is available when the user is requested for the service and it is available 24/7.The portal has a very low failure rate.

**3.13 Maintainability**

Very little maintenance should be required for this setup. An initial configuration will be the only portal required interaction after the portal 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 portal’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.

**3.14 Portability**

This portal should have the ability that, once it is together, the entire portal should be able to be physically moved to any location. Code and program portability should be possible between kernel-recompiled Linux distributions. For everything to work properly, all components should be compiled from source.

**3.15 Reusability**

The code written and the components used should have the ability to be reused with no problems. Should time allow, and detailed instructions are written on how to create this project, everything will be completely reusable to anyone.

**3.16 Application compatibility**

The Doctor Patient Management Portal is designed as an independent portal. As it is not connected to any other components or interfaces, application compatibility is not a concern.

**3.17 Resource utilization**

The Doctor Patient Management Portal uses very limited resources.

**3.18 Major Classes**

There are a total of seven major classes: User Login, User Profile, Doctor , Prescription, Schedule, and Appointment

**For User registration**:

* The patient and doctor registration has to be done and the registered data will be stored in patient and doctor database respectively.
* For login, the details from the doctor and admin should be authenticated and the details should match with the data file stored in database.

**New User:**

The user registration is to be created for doctors and patients and the data of the registered user is stored in doctor and patient database in a file. For login, the details from the doctor get compared with the data file stored in doctor database.

**Book Appointment:**

Once the registration is completed the patient has to choose the specialization, select the doctor and slot as per his choice.

**View Appointment:**

After registered or logged in, the patient can see his/her appointment details, doctor can see his /her patient appointment details and also the history of the patient if he/she had already received treatment and admin can see all the appointment details.

**Modifying Information:**

The doctor can only change his/her basic information .The admin can add/edit/delete the details of both doctor and patient.

**View Details:**

The patient can view his/her details. The doctor can view his/her details and the details of patients he treated and the admin can view both the details of doctor and patient.

**4.   Detailed Portal Design**

The portal design is as follows –

**4.1.  Design Descriptions**

Doctor Patient Management Portal is based on the concept of making patient appointments easier. The Protal helps to reduce the problems that occur when using a manual portal and helps patients to skip endless queues. Some hospitals provide the opportunity to make appointments by placing the call, but sometimes these calls are left unattended. The proposed portal will overcome all these drawbacks of the existing portal. The proposed portal has many advantages in that it stores all the information regarding patients' details, patient profiles, prescriptions, etc. Users can enter their details, update their profile and they can select doctors to make appointments. Other than that, the portal is user- friendly and it can be helpful to manage their appointments. The portal also helps to avoid duplicate appointments. The users can view available doctors and their timings and can make appointments accordingly. Users also get an option to cancel their appointments, view their upcoming appointments while past appointments are deleted automatically. One of the main advantages is that users are given an option to add their medical conditions if they want to. The protal allows the doctors to log in and can view their upcoming appointments, patient's case history, patient details, add prescription, book slot for patient’s revisit etc. The protal also has an administrator section that allows the administrator to manage the whole portal i.e. , he can add/remove appointments, doctors, and departments and search for appointments. Thus enabling doctors and hospital assistants to manage patient records and appointments. our protal also consists of organ donor module. this module allows for organ donation registration as well as organ search. the module is designed to help urgent organ requirements through easy/instant search.

**4.2 Design and Implementation Constraints**

The portal is built using the C language.

**4.3  User Interface**

     File portal interface.

**5. Demo**

**Main Menu**

A screenshot of a computer

Description automatically generated

**Patient’s Menu**

Text

Description automatically generated

**Doctor's Menu**

Graphical user interface

Description automatically generated with low confidence

**Admin’s Menu**

Text

Description automatically generated