

**Efficient Doctor Patient Management Portal**

High Level Design & Low Level Design

**Document Control :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Efficient Doctor Patient Management Portal** | | | | | | | | |
| Guided by-  **Mr. Sankar** |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| December 07 2022 | 1.0 | B. Navya Sree  P. Gopika  P. Lasya  N. Harshitha  G. Sireesha |  | | | |  | |

**INDEX**

1. Introduction ------------------------------------------------ 4

1.1 Intended audience ------------------------------------------------ 4

1.2 Project purpose ------------------------------------------------ 4

1.3 Key project objective ------------------------------------------------ 4

1.4 Project scope ------------------------------------------------ 4

2. Design overview ------------------------------------------------ 4

2.1 Design objectives ------------------------------------------------ 8

2.2 Error Detection / Exceptional Handling -------------------------------------------- 8

2.3 Performance ------------------------------------------------ 8

2.4 Maintenance ------------------------------------------------8

3. System architecture ------------------------------------------------ 9

3.1 Doctor -----------------------------------------------9

3.2 Patient -----------------------------------------------9

3.3 Admin -----------------------------------------------10

3.3.1 Doctor Record maintenance ----------------------------------------------10

3.3.2 Patient Record Maintenance ----------------------------------------------10

3.4 File Handling -----------------------------------------------10

3.4.1 Doctor Data -----------------------------------------------10

3.4.2 Patient Data ---------------------------------------------11

4. Detailed system design -----------------------------------------------12

4.1 Flowchart of main application ----------------------------------------------12

4.2 DFD Level 0 Diagram ------------------------------------------------13

4.4 DFD Level 1 Diagram ----------------------------------------------13

5. Tools Report ---------------------------------------------14

5.1 Splint Report ----------------------------------------------14

5.2 Valgrind Report ------------------------------------------15

6. Testing

6.1 Unit Testing ----------------------------------------------16

6.2 Integration Testing —-------------------------------------------17

7. Environmental Description -----------------------------------------------18

# 1: Introduction

The purpose of this project is to show the requirements for the Food Waste Management application, in which either NGO request the hotel/restaurant or the hotel/restaurant requests the NGO for the left-over food which is to be supplied for the needy.

**1.1 Intended Audience:**

This document is intended to be read by, Doctor, Patient and admin.

**1.2 Project 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 .

.

**1.3 Key Project Objectives:**

1. Allows Doctor patient and admin to register.
2. Patient to book appointmnet.
3. Doctor can view and update the patient history.
4. File Maintenance by admin.
5. Admin can modify the data.

**1.4 Project 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

**2. Design Overview:**

Docto patient Management comprises of the following modules:

|  |  |
| --- | --- |
| Name of the module | Doctor Registration |
| Handled by | Bharapaneedi Navya Sree |
|  |  |
| Description | This feature allows the NGO to add a new record. It asks for the NGO registration id, name of NGO, number of beneficiaries, established year, office address and the telephone number |

|  |  |
| --- | --- |
| Name of the Module | Doctor edit |
| Handled by | Bharapaneedi Navya Sree |
| Description | This feature allows the doctor to edit basic details. |

|  |  |
| --- | --- |
| Name of the Module | Doctor View |
| Handled by | Bharapaneedi Navya Sree |
| Description | This feauture displays basic details of doctor. |

|  |  |
| --- | --- |
| Name of the Module | Doctor Login |
| Handled by | P.Gopika |
| Description | This feature allows the NGO to request the left-over food from the hotel / restaurant. If the request is accepted by the hotel / restaurant the conformation slip is generated. If not the request will be forwarded to admin. |

|  |  |
| --- | --- |
| Name of the Module | Patient registration |
| Handled by | Pandeti Lasya |
| Description | This feature allows the hotel to add the hotel/restaurant record. It asks for the hotel/restaurant registration id, name,  address and contact details. |

|  |  |
| --- | --- |
| Name of the Module | Patient Login |
| Handled by | P.Gopika |
| Description | This feature allows the Patient to login into the portal. |

|  |  |
| --- | --- |
| Name of the Module | Doctor Slot selection |
| Handled by | P. Lasya |
| Description | This feature doctor to select slot that he is available. |

|  |  |
| --- | --- |
| Name of the Module | Doctor list view |
| Handled by | G. Sireesha |
| Description | This feature displays the list of available doctors with specialization |

|  |  |
| --- | --- |
| Name of the Module | Slot selection |
| Handled by | G. Sireesha |
| Description | This feature allows patient to select in which slot he wants to visit the doctor. |

|  |  |
| --- | --- |
| Name of the Module | Docto Edit |
| Handled by | G. Sireesha |
| Description | This feature allows doctor to edit his basic details like name, mobile no,age. |

|  |  |
| --- | --- |
| Name of the Module | Patient Edit |
| Handled by | P. Lasya |
| Description | This feature allows doctor to edit patients medical information |

|  |  |
| --- | --- |
| Name of the Module | Patient View |
| Handled by | P. Gopika |
| Description | This feature displays the patient list to doctor |

|  |  |
| --- | --- |
| Name of the Module | Doctor Add |
| Handled by | B. Navya Sree |
| Description | This feature adds the new doctor record with details like mci licence id, age, name, gender, mobile number and password |

|  |  |
| --- | --- |
| Name of the Module | Doctor Edit |
| Handled by | P.Gopika |
| Description | This feature allows admin to edit the doctor details. |

|  |  |
| --- | --- |
| Name of the Module | Doctor Delete |
| Handled by | P. Lasya |
| Description | This feature allows admin to delete doctor data from the portal |

|  |  |
| --- | --- |
| Name of the Module | Patient edit |
| Handled by | G. Sireesha |
| Description | This feature allows admin to edit or delete the patient details |

|  |  |
| --- | --- |
| Name of the Module | Admin Login |
| Handled by | N.Harshitha |
| Description | This feature allows admin to login into the portal to see doctor and patient information |

|  |  |
| --- | --- |
| Name of the Module | Doctor List |
| Handled by | N. Harshitha |
| Description | This feature will give a report on the number  Of doctors available in the portal so the admin can edit the details- of the doctor |

|  |  |
| --- | --- |
| Name of the Module | Doctor file to list |
| Handled by | N. Harshitha |
| Description | This feature allows to write Doctor records to the list. |

|  |  |
| --- | --- |
| Name of the Module | Doctor List to File |
| Handled by | N.Harshitha |
| Description | This feature allows to read Doctor records to the list. |

|  |  |
| --- | --- |
| Name of the Module | Patient File to list |
| Handled by | N. Harshitha |
| Description | This feature allows to write Patient records to the list. |

|  |  |
| --- | --- |
| Name of the Module | Patient List to File |
| Handled by | N. Harshitha |
| Description | This feature allows to read patient records to the  list. |

**2.1 Design Objectives:**

Food Waste Management Application communicates among NGO’s, Hotels and admin for the supply of left-over food to the needy.

The main design features include four major parts:

- The architecture,

- The user interface design,

- The files,

- Process relation,

- Automation .

**2.2 Error Detection / Exceptional Handling:**

If the Doctor or Patient wants to register with the existed user ID then it displays “User already Exists”. If the patient wants to edit doctor details it displays “Permission Denied”.

**2.3 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.

**2.4 Maintenance:**

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. SYSTEM ARCHITECTURE: -**

**3.1 Doctor:**

**3.1.1 : doctor\_registration:** This feature allows to add a new Doctor record. I asks for the Doctor MCI licence id, name of Doctor, Age, gender, weight, Mobile number and password.

**3.1.2 : doctor\_slot\_selection:** This feature allows the Doctor to select slots that he is available.

**3.1.3doctor\_file\_maintenance:** This feature displays the details of Doctor.

**3.1.4: view\_patient\_history :** This feauture allows doctor to view patient medical history

## 3.2 Patient:

**3.2.1 :patient\_Registration**: This feature allows to add the new Patient record . It asks for the Aadhar Number , Name, Age, Gender, weight and contact details.

**3.2.2 :patient\_login:** This feature allows the Patient to login into the portal

**3.2.3 :book\_appointment:** This feature allows patient to book appointment.

## 3.3 Admin:

* **Admin\_login:** This feature allows admin to login with id and password
* **Edit\_doctor\_info:** This feature allows admin to add, modify or delete the doctor details.
* **Edit\_patient\_info:** This feature allows admin to add or delete patient data.

**3.4 : File Handling:**

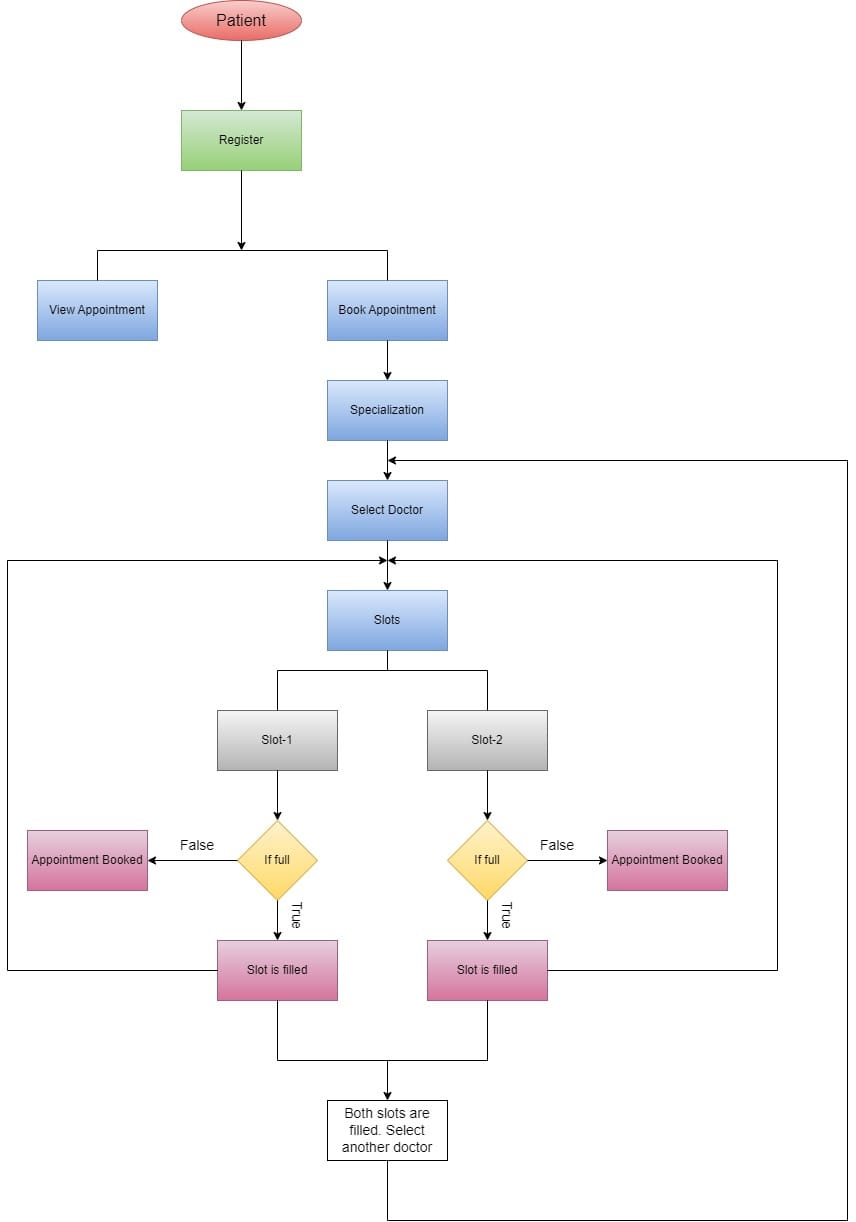
**3.4.1 : Doctor File Handling:**

* **Doctor\_file\_to \_list:** This feature allows to write Doctor records to the list.
* **Doctor\_list\_to\_file:** This feature allows to read Doctor records from the list.

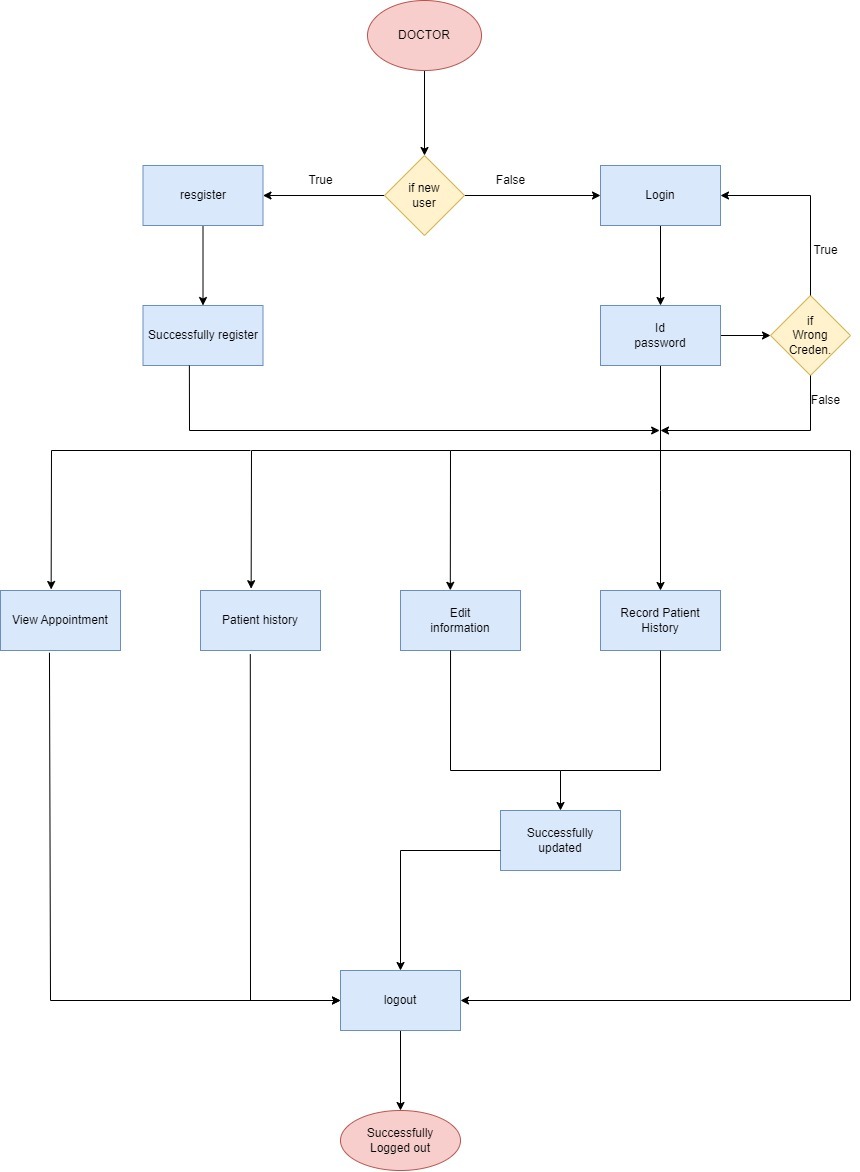
**3.4.2 : Patient File Handling:**

* **Patient\_file\_to\_list:** This feature allows to write Patient records to the list.
* **Patient\_list\_to\_file:** This feature allows to read Patient records from the list.

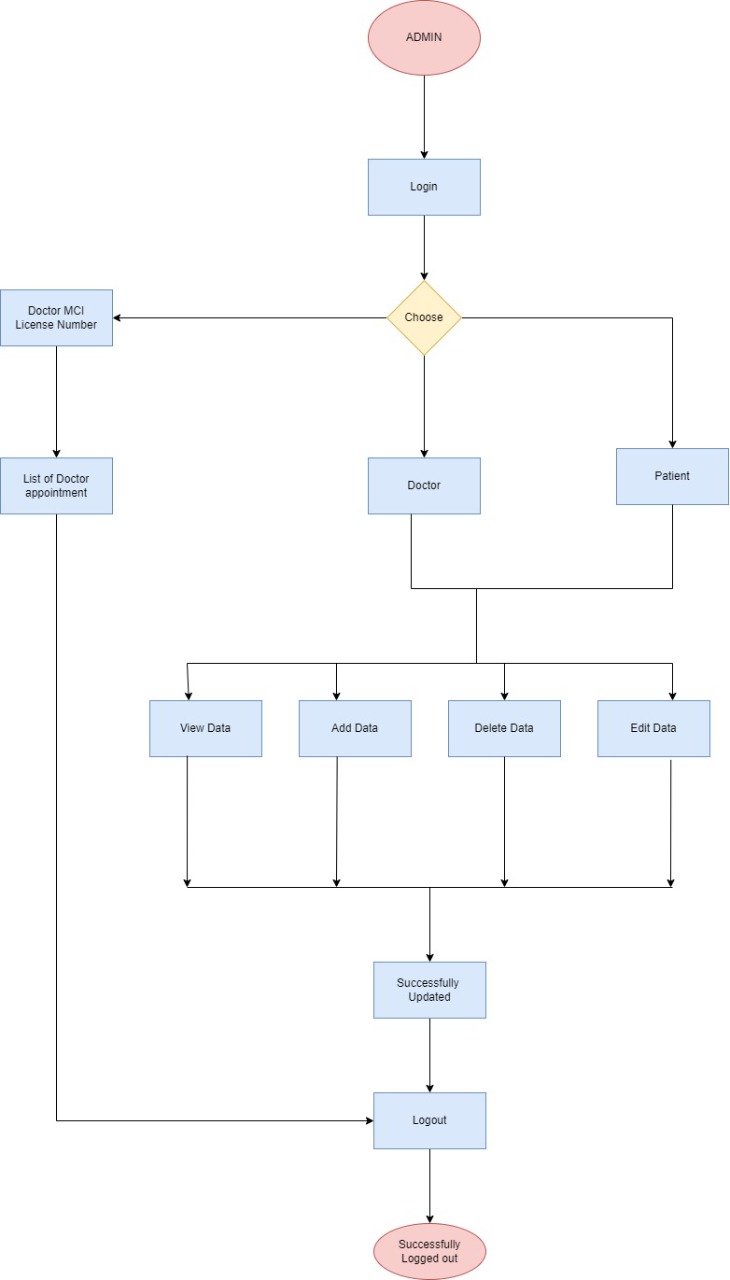
**4. Detailed System Design**



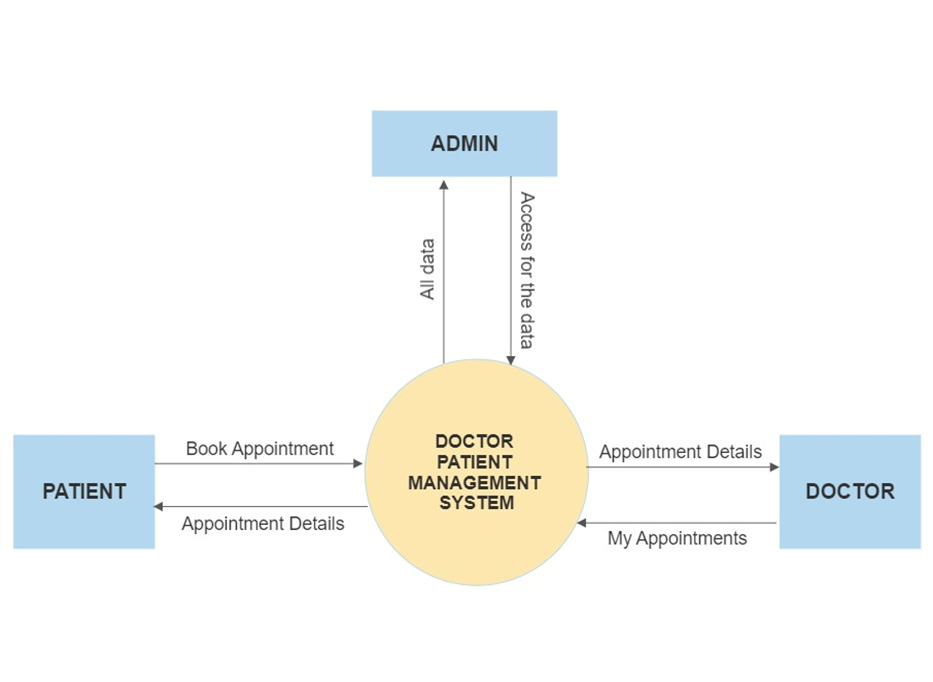
4.1Patient Flow Chart



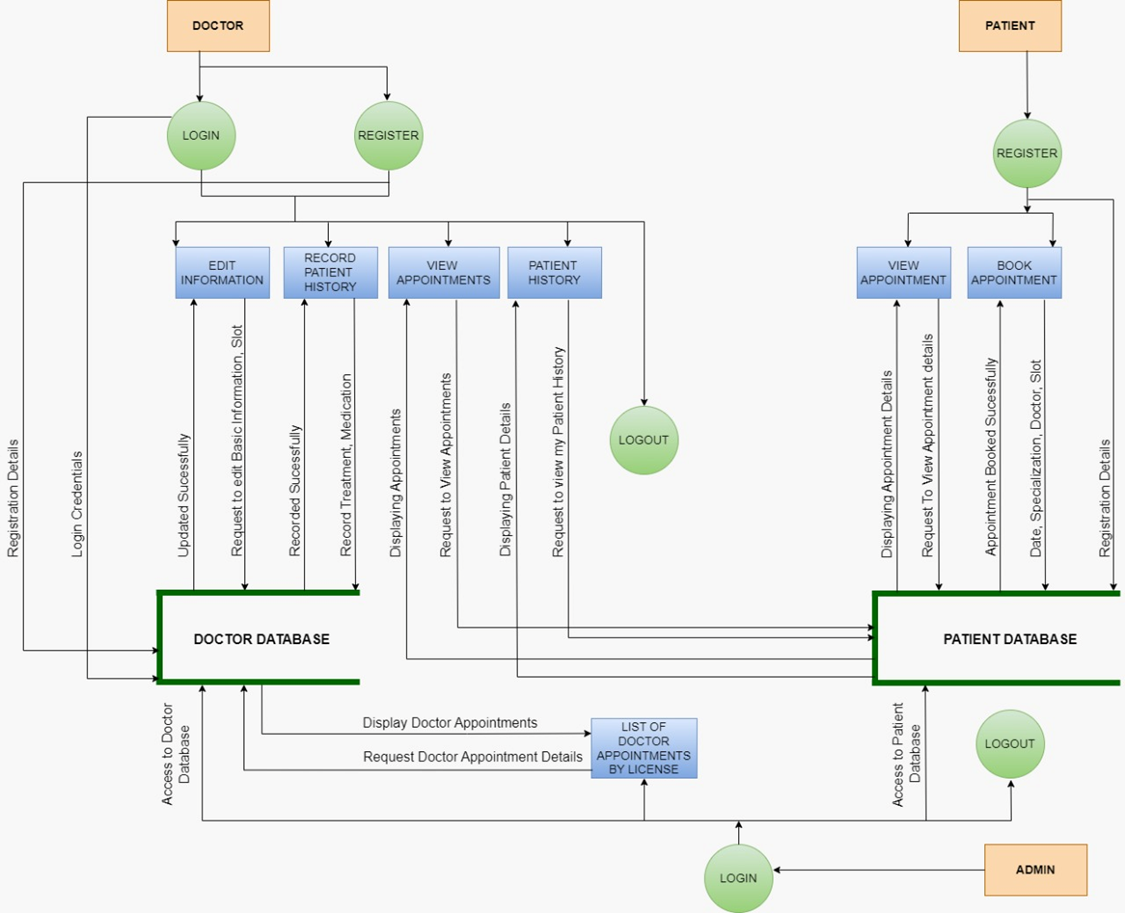
4.2 Doctor flow chart



4.3 Admin Flow Chart

****

**4.2 DFD LEVEL 0 DIAGRAM**

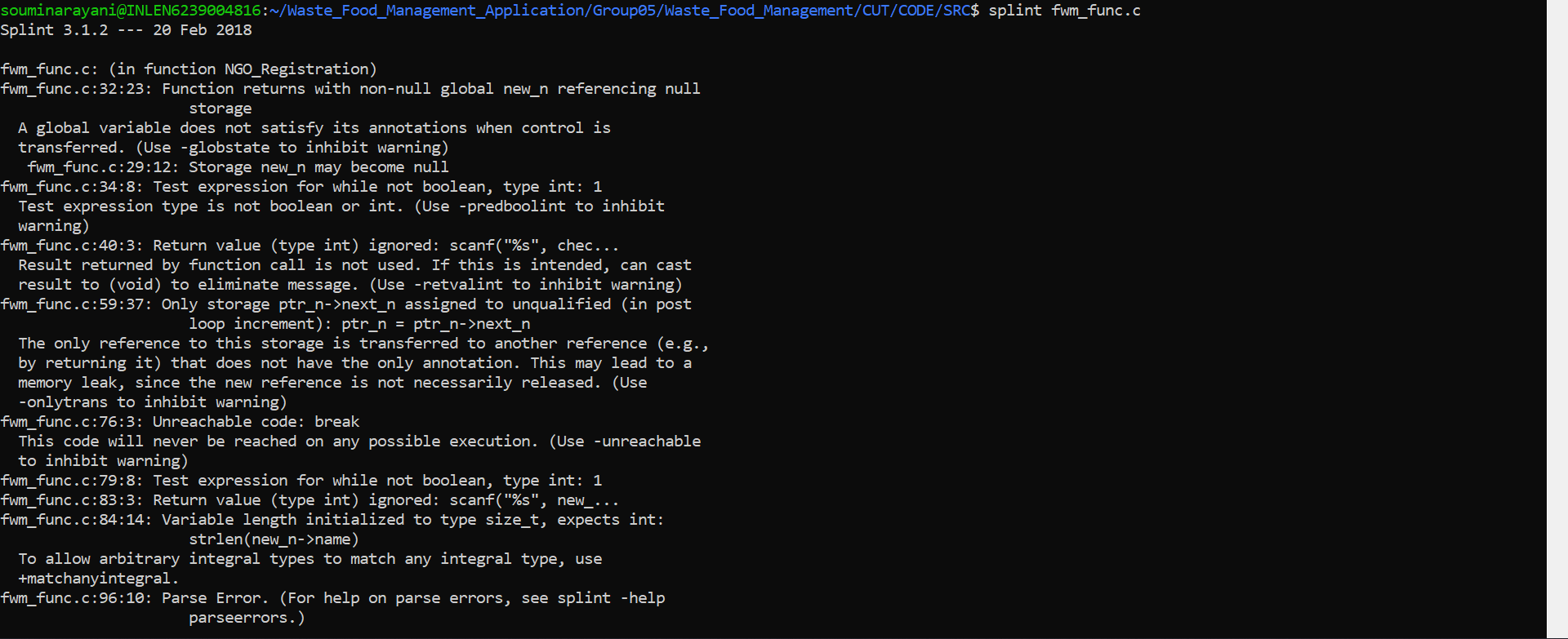
****

4.3 **DFD LEVEL 1 DIAGRAM**

**5 : Tools Report**

**5 .1 :Splint Tool Report**

fwm\_func.c: (in function NGO\_Registration)  
fwm\_func.c:32:23: Function returns with non-null global new\_n referencing null storage  
  A global variable does not satisfy its annotations when control is  
  transferred. (Use -globstate to inhibit warning)  
  fwm\_func.c:29:12: Storage new\_n may become null  
 fwm\_func.c:34:8: Test expression for while not boolean, type int: 1  
  Test expression type is not boolean or int. (Use -predboolint to inhibit  
  warning)  
 fwm\_func.c:40:3: Return value (type int) ignored: scanf("%s", chec...  
   Result returned by function call is not used. If this is intended, can cast  
   result to (void) to eliminate message. (Use -retvalint to inhibit warning)  
 fwm\_func.c:59:37: Only storage ptr\_n->next\_n assigned to unqualified (in post  
                     loop increment): ptr\_n = ptr\_n->next\_n  
  The only reference to this storage is transferred to another reference (e.g.,  
  by returning it) that does not have the only annotation. This may lead to a  
  memory leak, since the new reference is not necessarily released. (Use  
  -onlytrans to inhibit warning)  
 fwm\_func.c:76:3: Unreachable code: break  
  This code will never be reached on any possible execution. (Use -unreachable  
  to inhibit warning)  
 fwm\_func.c:79:8: Test expression for while not boolean, type int: 1  
 fwm\_func.c:83:3: Return value (type int) ignored: scanf("%s", new\_...  
 fwm\_func.c:84:14: Variable length initialized to type size\_t, expects int:  
                     strlen(new\_n->name)  
  To allow arbitrary integral types to match any integral type, use  
  +matchanyintegral.  
 fwm\_func.c:96:10: Parse Error. (For help on parse errors, see splint -help  
                     parseerrors.)



1. **Testing**

**6.1 : Unit Testing :**

UT\_01 : NGO\_Registration

A screenshot of a computer

Description automatically generated with medium confidence

UT\_02 : NGO\_Edit

Text

Description automatically generated

UT\_03 : NGO\_View

A screenshot of a computer screen

Description automatically generated

**6.2 : Integrating Testing :**

IT\_01 : NGO

A screenshot of a computer

Description automatically generated with medium confidence

IT\_02 : Hotel

A screenshot of a computer

Description automatically generated with medium confidence

IT\_03 : Admin

A screenshot of a computer

Description automatically generated with medium confidence

**7.Environment Description:**

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

**7.2 Language Support:** English

**7.3 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

**7.4 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

**7.4.1 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.

**7.4.2 Application Server Disk Space:**

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.

**7.4.3 Database Server Disk Space:**

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.

**7.4.4 Integration Requirements:**

1. Language: C

2. Tools: Splint , Valgrind

3. Complier: gcc

4. Linux Environment

**7.4.5 Jobs:**

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

**7.4.6 Network:** End to End

**7.5 Configuration:**

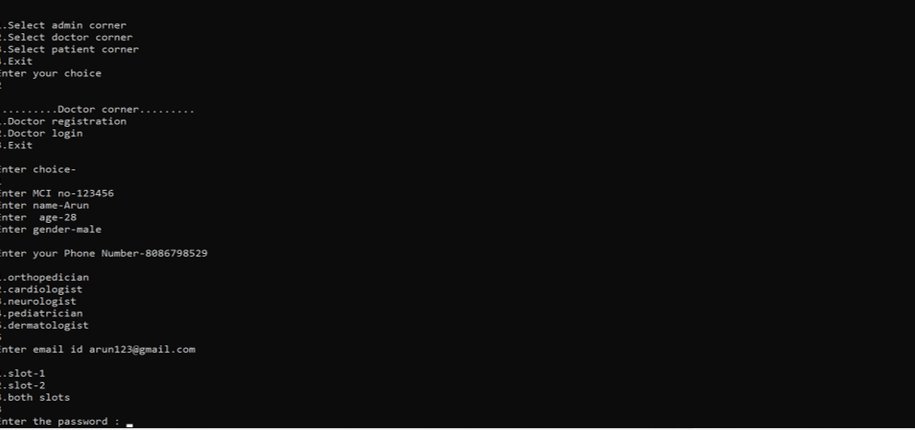
**7.5.1: Operating System**: Linux environment

Output:

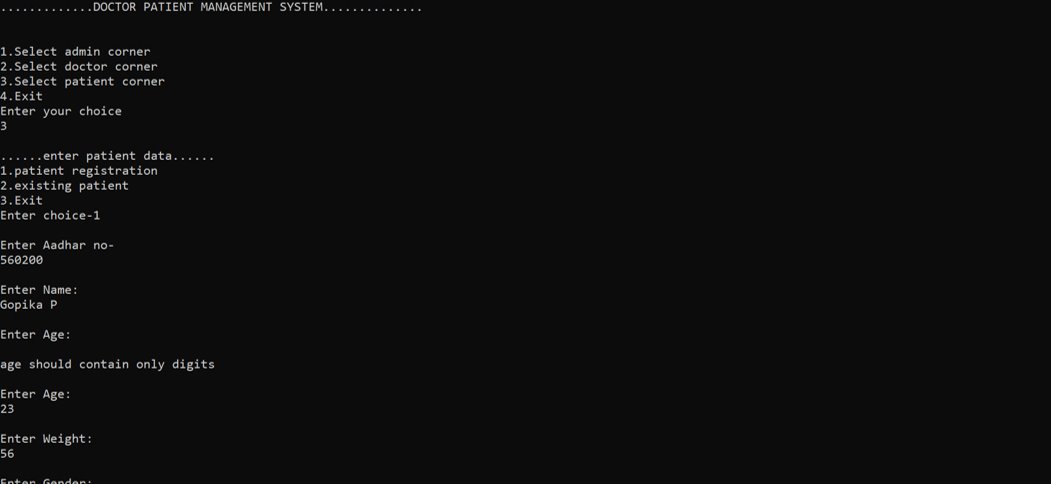
Main window:



Doctor registration:



Patient registration:



Slot booking:



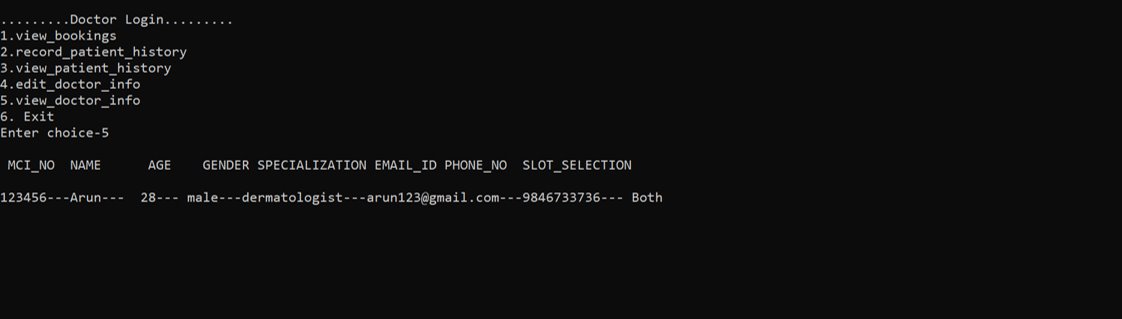
Record Patient medical history:



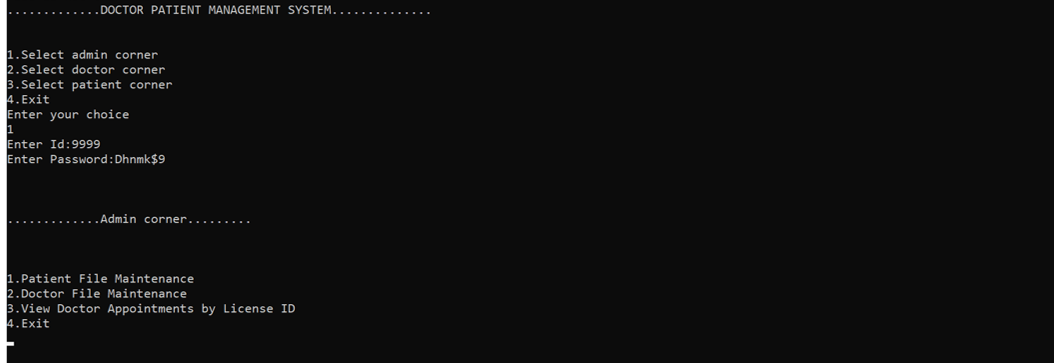
View patient history:



View doctor’s profile:



Admin portal:



View doctor’s from admin portal:

