Doctor Appointment Management System

(DAMS)

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
21/10/2022	V0.1	Group 1	Initial draft	

TEAM MEMBERS

EMP Number:	Name
46275106	ARUNABHA SARKAR
46274550	AMISHA BHARTI
46274781	APARNA
46274780	ABBURI HARSHITHA CHOWDARY
46274551	ALOK KUMAR SAHU

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………………………………………………………………….…...8

3.4. Data Flow Diagram…………………………………………….………..…...8

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……………………………………….………………………...12

3.13. Reliability……………………………...…………………………….12

3.14. Maintainability……………………………...…………………………...12

3.15. Portability……………………………………………...………………….12

3.16. Reusability…………………………………………………..………..…...12

3.17. Application compatibility……………………..………………..…….…...13

3.18. Resource utilization……………………………………..…………………….13

3.19. Major Classes……………………………………...……………..………13

4. DETAILED SYSTEM DESIGN………………………..…...…16

4.1. Design Descriptions…………………………………………...……………16

4.2. E-R Diagram……………………………………………………...……….17

4.3. Use Case Diagram…………………………………………………..…...18

4.4. Design and Implementation Constraints………………….……………..…...18

4.5. User Interface……………………...…………………..…………...………18

5. SECURITY……………………………………………..................19

6. DEMO…………………………………………………..........…19

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 system that will help doctors in their work and will also help patients to book doctor appointments and view medical progress. The system 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 system 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 system.
2.Programmin Language - C language
3.Pictorial representations – Canva

2.3 General Constraints

The Doctor Appointment Management System 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 system is that it will work with a single user at a time .

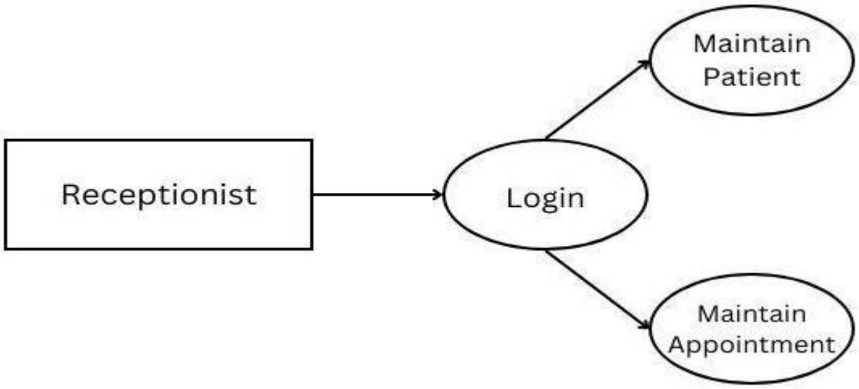
3. Design Details

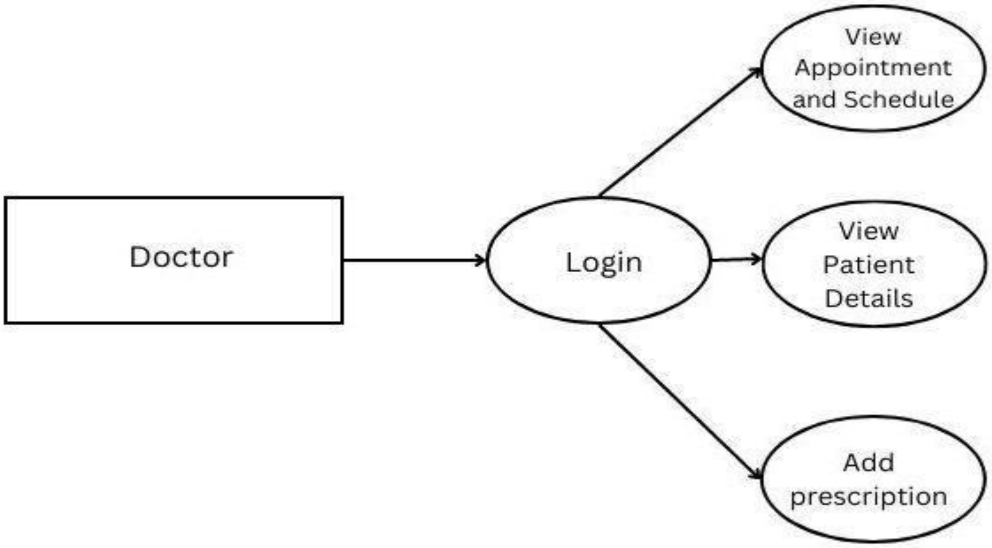
3.1 Main Design Features

The main design features include four major parts:
- The architecture,
- The user interface design,
- The files,
- Process relation,
- Automation .

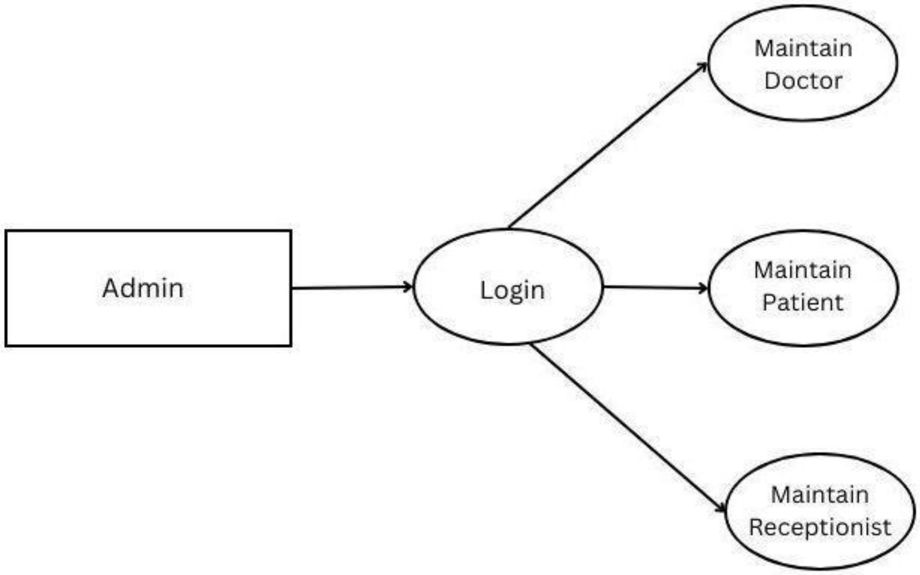
In order to make these designs easier to understand, the design has been illustrated in attached diagrams (Use Case, Data flow diagrams).

3.2 Application Architecture





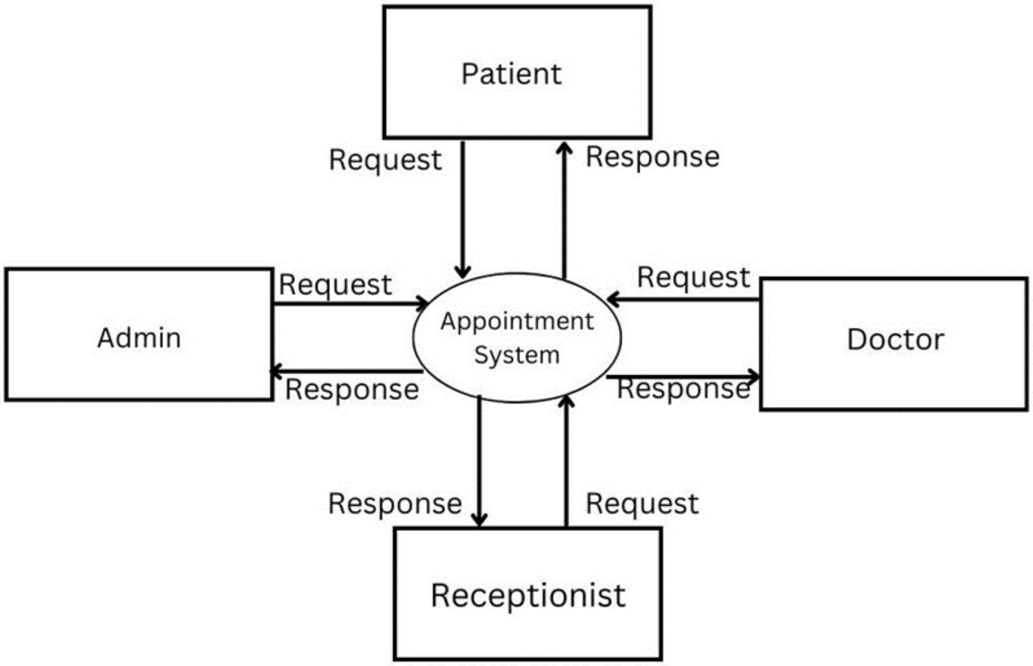
Printed copies are current on date of printing only - 10/27/2022. Always refer to the electronic version for the current release.



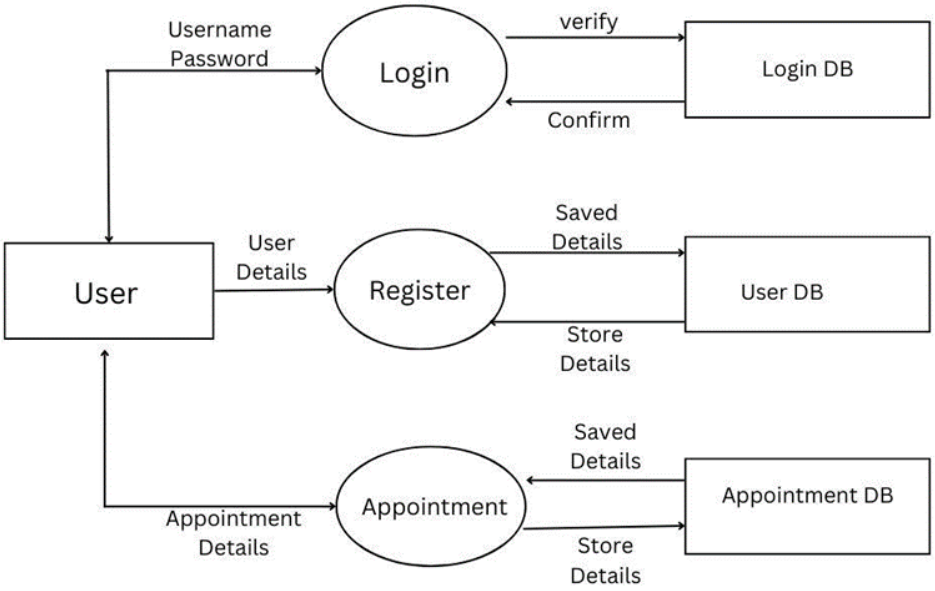
3.3 Standards

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

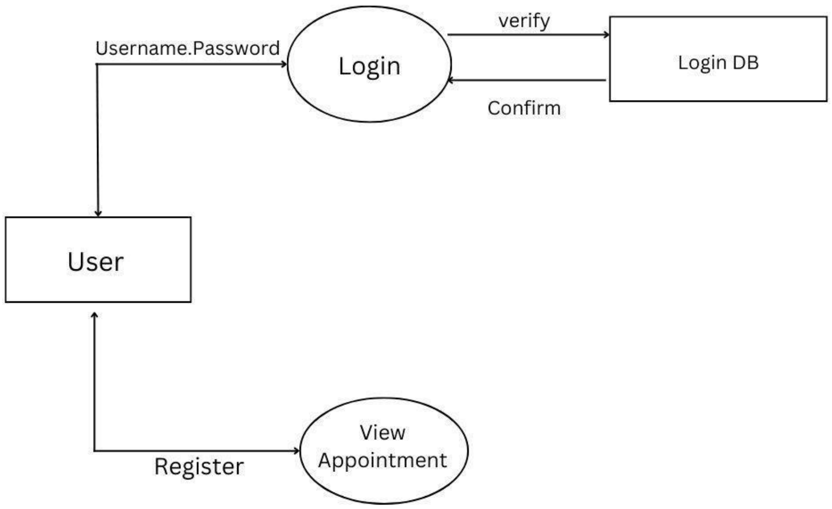
3.4 Data Flow Diagram



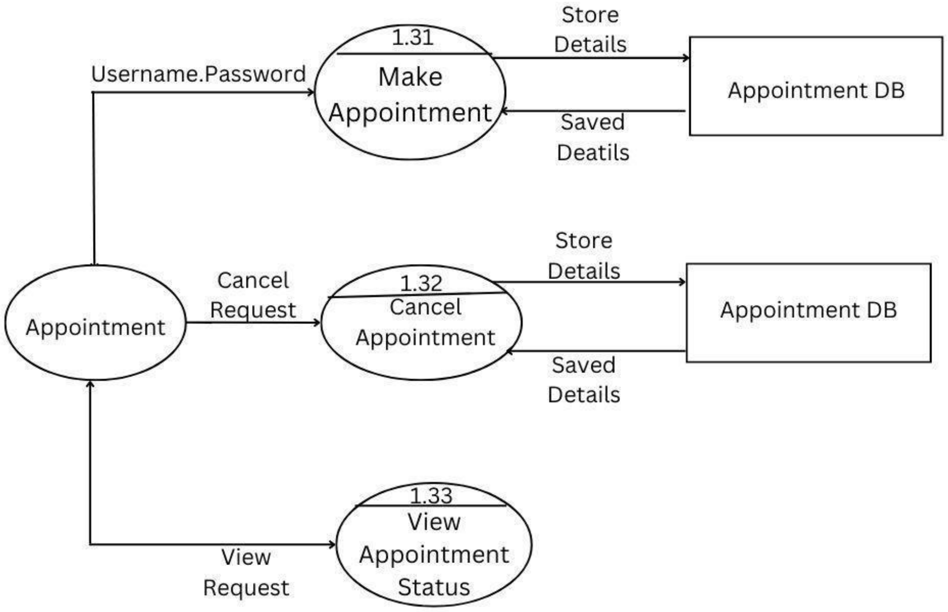
LEVEL 1 DFD FOR PATIENT



DFD FOR DOCTOR



LEVEL 1.30 DFD FOR APPOINTMENT



3.5 Files

The DAMS 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 system 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 system 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 system will work on the customer's terminal and the performance depends upon the hardware component of the customer’s system.

3.12 Security

The customer’s terminal window shall never display a customer’s password. It shall always be echoed with special characters representing typed characters. The system’s back-end files shall never display a customer’s password. The customer’s password may be reset but never shown.

3.13 Reliability

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

3.14 Maintainability

Very little maintenance should 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.

3.15 Portability

This system should have the ability that, once it is together, the entire system 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.

3.16 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.

3.17 Application compatibility

The DAMS is designed as an independent system. As it is not connected to any other components or interfaces, application compatibility is not a concern.

3.18 Resource utilization

The DAMS uses very limited resources.

3.19 Major Classes

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

Actors	Description

User	● On registration and Login The system allows the user to create a profile and set his credential. The system authenticates user credentials to view the profile. The system allows the user to reset the password by selecting the forgot password option. ● Allows medical profile creation The system allows the user to create a medical profile which includes basic information regarding the health condition of the user. ● Provides provision for patients to update their medical Profile The system allows users to make changes on their medical profile ● Facility to book an appointment The system will display the available doctors and their time to the user on selecting the department. The system allows the users to book an appointment according to their respective disease. ● Later on by adding their details an appointment is fixed. The system then fixes appointments with the available doctor.

Doctor	• Login • Doctor can login to the system, Provide doctors with their schedules The system will show the list of their daily appointment schedules and provides more details on each appointment on request. • Provide provision for doctors to add a diagnoses , add prescriptions of their patient The system allows doctors to add diagnoses of their patients. Also doctors can add prescriptions for the patient after each appointment. • Provide provision for doctors to update the appointment status. The system allows the doctor to select and update the appointment status . • Doctors can book the slot for patients for a revisit. The system allows the doctor to book slots for patients who are suggested for a revisit after a few days. • Provide provision for doctors to view appointment history The system allows the doctor to view the past appointment history. • Provide provision for doctors to search a patient The system allows the doctor to search a patient by their patient ID.

Admin	● Login Admin can login to the system. ● Admin can add , update and delete doctors. The system allows the admin to add new doctors to the doctors list as well as they can remove the doctor from the existing doctors list. Also, admin can make changes to doctor's information if needed. ● Provides provision for admin to add, update and delete schedules of their doctors .The system allows the admin to add schedules for their doctors as well as update the existing schedule. Also, the admin can remove or cancel the schedule of doctors in certain situation like if the doctor is not available.

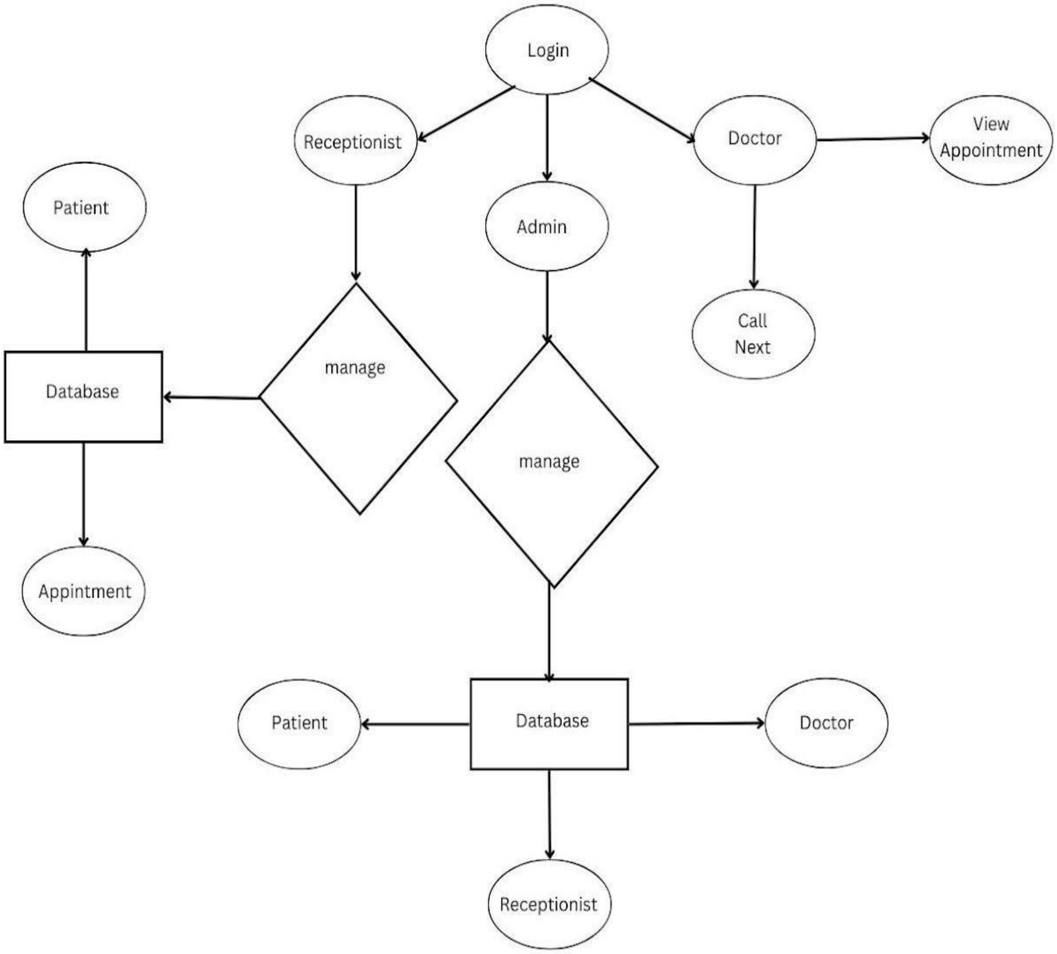
4. Detailed System Design

The system design is as follows –

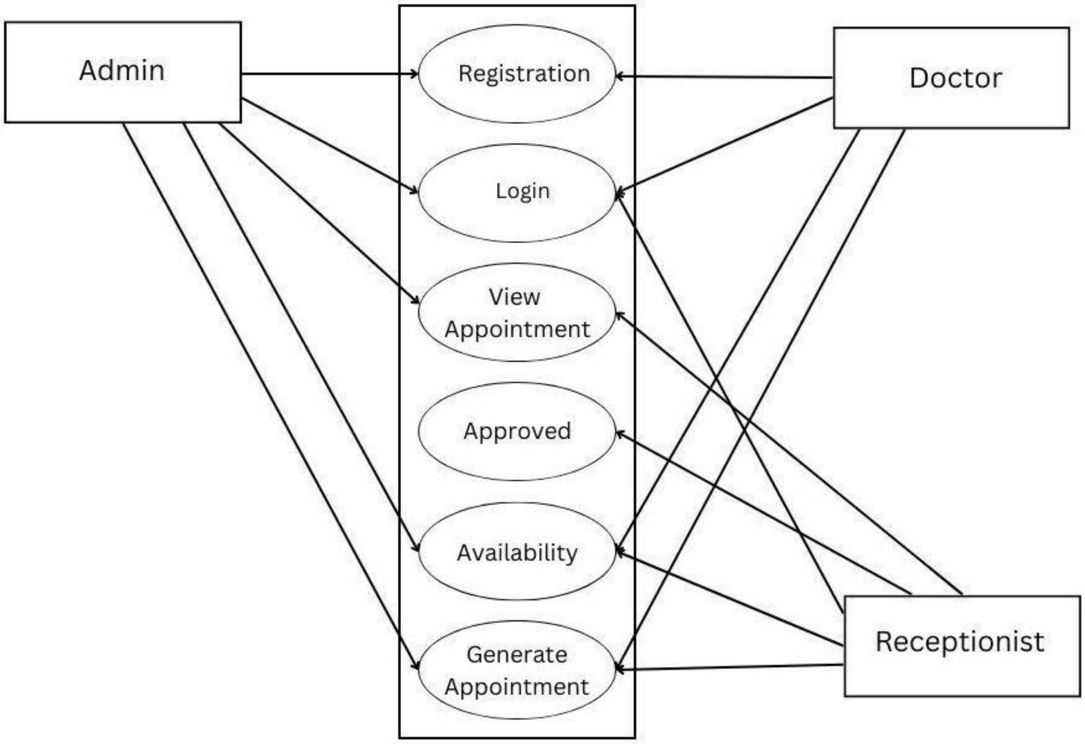
4.1. Design Descriptions

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 queues. 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. Users can enter their details, update their profile and they can select doctors to make appointments. Other than that, the system is user- friendly and it can be helpful to manage their appointments. The system 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 system 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 system also has an administrator section that allows the administrator to manage the whole system 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.

4.2. E-R Diagram



4.3. Use Case Diagram



4.4 Design and Implementation Constraints

The system is built using the C language.

4.5 User Interface

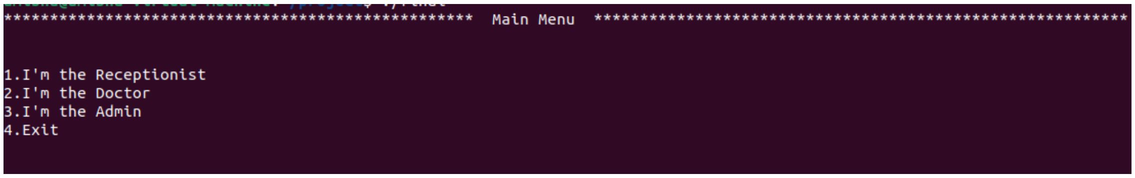
File system interface.

5. Security

The customer’s terminal window shall never display a customer’s password. It shall always be made with special characters representing typed characters. The system’s back-end files shall never display a customer’s password. The customer’s password may be reset but never shown.

6. Demo

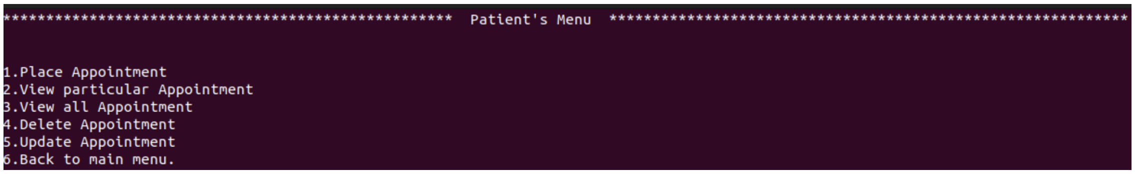
Main Menu



```
***************************************************  Main Menu  ************************************************************


1.I'm the Receptionist
2.I'm the Doctor
3.I'm the Admin
4.Exit
```

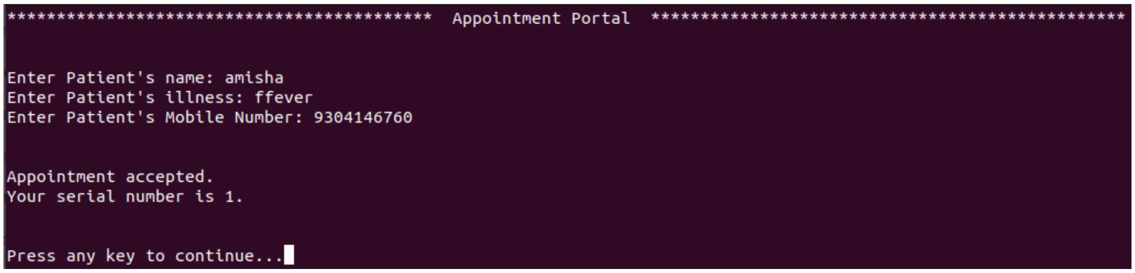
Patient’s Menu



```
**************************************************  Patient's Menu  ***********************************************************


1.Place Appointment
2.View particular Appointment
3.View all Appointment
4.Delete Appointment
5.Update Appointment
6.Back to main menu.
```

Appointment Portal



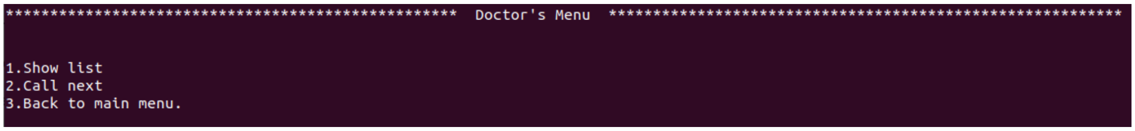
```
*******************************************  Appointment Portal  ************************************************


Enter Patient's name: amisha
Enter Patient's illness: ffever
Enter Patient's Mobile Number: 9304146760


Appointment accepted.
Your serial number is 1.


Press any key to continue...
```

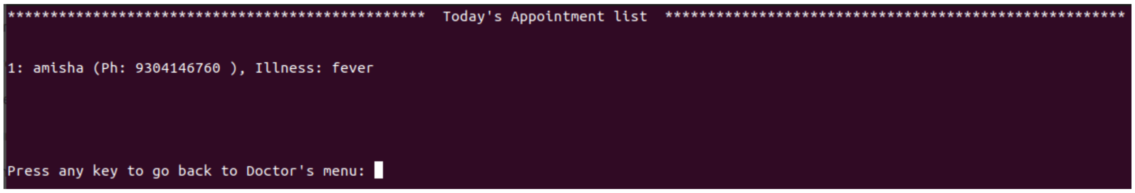
Doctor's Menu



```
**************************************************  Doctor's Menu  **********************************************************


1.Show list
2.Call next
3.Back to main menu.
```

Today's Appointment List



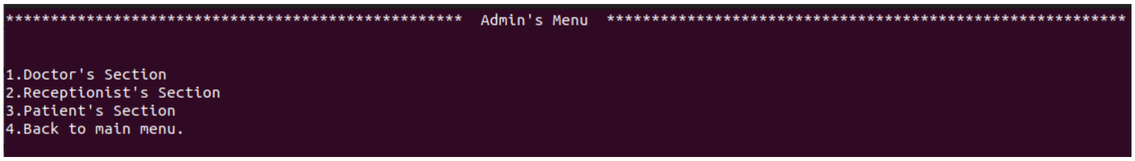
```
*************************************************  Today's Appointment list  ******************************************************


1: amisha (Ph: 9304146760 ), Illness: fever




Press any key to go back to Doctor's menu: 
```

Admin’s Menu



```
******************************************************  Admin's Menu  *************************************************************


1.Doctor's Section
2.Receptionist's Section
3.Patient's Section
4.Back to main menu.
```