LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

Software Requirements Specification (SRS)

Automate all the functional elements in Sh. Baldev Raj Mittal hospital

Submitted To

Uminder Kaur

PREPARED BY		
STUDENT NAME	**ROLL NUMBER**	**REGISTRATION NO.**
PRITI KUMARI	RD2217A4	12212485

Lovely Professional University, Punjab(India)

Contents

1.Introduction

1.1 Purpose

The purpose of this product is to bring together all the hospitals, doctors, staffs, patients and other respective parties related to medical care under a single system to facilitates interlinking between different parties and to facilitates more effective service to consumers. The application aims to maintain a global database to all parties to provide better service. The application is being developed taking into consideration the consumers who through this system will have more options access and hospitals who can manage their daily needs efficiently.

1.2 Product Scope

Currently Sh. Baldev Raj Mittal Hospital is using a manual system to handle the hospital process. When patients arrive, they make an appointment at the reception to consult a doctor. They are being recorded in a file. Then again, the patients diagnosed symptoms related disease details, ward details and other necessary details are being recorded and those files are being stored in special locations. Calculations of bills and inventory are done manually.

As the current system is a file based one, management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also, could be misplaced by losing data and information.

Limited storage space of the files is another issue that they currently face when the management is manually done.

If we want to check a previous record of a patient or other detail. Management will be in a great problem. It's a tough and time taking process to search for a record in a file.

We plan to overcome the above-mentioned problems through a standalone application, to manage the major functions of the Hospital System.

The hospital management system we are going to implement will overcome all the basic process done in the hospital. It would handle

- ✓ Employee and Salary management
- ✓ Patient and "Zumba exercise" management
- ✓ Theatre and ward Management, Laboratory managements
- ✓ Transport managements
- ✓ Pharmacy Managements
- ✓ OPD managements and emergency managements.

Implementing the Employee & Salary Management system we record Attendance, shifting of employees, their holidays and consulting doctors' schedules.

The proposed system for Mini-theatre & Ward Management records details of surgeons, in- patients who are assigned for Wards, different ward details and surgery details. The

pharmaceuticals used within the theatre are managed as well. Food menus for the patients according to their diseases based on wards is systemized too. Surgery reports, Ward progress reports, In-ward patient progress details are generated and history can be tracked too.

The Vehicle & Transport Management system handles all the data on ambulance transport. It manages the time slots of ambulances, driver's and employee details of transport section and provides bill generating facility. And reliable time slot management provides the facility of checking the availability of the ambulances whenever required, and decide about a possible time they can fulfil a request.

The system developing for Emergency Treatment & Equipment Management automate the current processes of patient registering and propose a better way to keep records of equipment and medicines related to the emergency treatment unit in a computer-based file system.

The Pharmacy Stock Management system is responsible for proper management of drug stocks, pop ups the notifications of expiry dates of stock items. This system allows the client to keep track of medicine stocks, notify the personals when the stock is running out of items and help the manager to reduce stock levels and eliminate stock waste.

The Lab Management System records sample collection details, keep track of lab resources and participate in lab reports conclusion generating. This increases the accuracy of report generating process and save a lot of time in manual handling of report details and improve the efficiency and the productivity of the organization.

In OPD unit, with the OPD and Consultation Management system, the manual doctors channelling details entering process has automated. So, the staff does not need to spend time on writing appointment records and updating them in files.

1.3 Overview

- ✓ This software Requirements Specification (SRS) is the requirements work product that formally specifies Hospital Management System (HMS).
- ✓ It includes the results of both business analysis and system analysis efforts Various techniques were used to elicit the requirements and we have identified your needs, analysed and refined them.
- ✓ The objective of this document therefore is to formally describe the system's high-level requirements including functional requirements, non-functional requirements and business rules and constraints. The detail structure of this document is organized as follow:

- ✓ Section 2 of this document provides an overview of the business domain that the proposed Hospital Management System (HMS) will support.

- ✓ These include a general description of the product, user characteristics, general constraints, and many assumptions for the system.
- ✓ This model demonstrates the development team's understanding of the business domain and serves to maximize the team's ability to build a system that truly does support the business.

Advantages

- The system automates the manual procedure of managing hospital activities.
- Doctors can view their patients' treatments records and details easily.
- It even generates an instant bill.
- The system is convenient and flexible to be used.
- It saves their time, efforts, money and resources.

Disadvantages

- Requires large database.
- The admin has to manually keep updating the information by entering the details in the system.
- Need internet connection.

2. Overall description

2.1 Product Perspective

This Hospital Patient Info Management System is a self-contained system that manages activities of the hospital. Due to improperly managed details medical centre faces quite a lot of difficulties in accessing past data as well as managing present data. The fully functional automated hospital management system which will be developed through this project will eliminate the disadvantages caused by the manual system by improving the reliability, efficiency and performance. The usage of a database to store patient, employee, stock details etc. will accommodate easy access, retrieval, and search and manipulation of data. The access limitations provided through access privilege levels will enhance the security of the system. The system will facilitate concurrent access and convenient management of activities of the medical centre.

2.1.1 System Interface

❖ **User Interfaces**

- This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interfaces.
- The protocol used shall be HTTP
- The port number will be 80.
- There shall be logical address of the system in IPv4 format.

❖ **Hardware Interface**

- Laptop/Desktop PC
- Purpose of this pc is to give information when Patients ask information about doctors, medicine available lab tests etc. To perform such Action, it needs very efficient computer otherwise due to that reason patients have to wait for a long time to get what they ask for.

▪ **Display Unit (LED/LCD Monitor/TV)**

This unit is for display the channel number when the patients come to see their consultants. It will avoid chaos. And also display Hospital welcome screen, video, information etc.

▪ **Laser Printer (B/W)**

Simply this device is for printing bills and view reports.

▪ **Wi-Fi router**

Wi-Fi router is used to for internetwork operations inside of a hospital and simply data transmission from pcs to sever.

❖ **Software Interfaces**

Developing end

- **C++** - C++ is secure and reliable. From laptops to datacentres, game console to scientific supercomputers, cell phones to the Internet.
- **VS Code** - IDE for C++ Developing.
- **MySQL server** - Database connectivity and management
- **Adobe Photoshop** - Logo and other designing such User interfaces

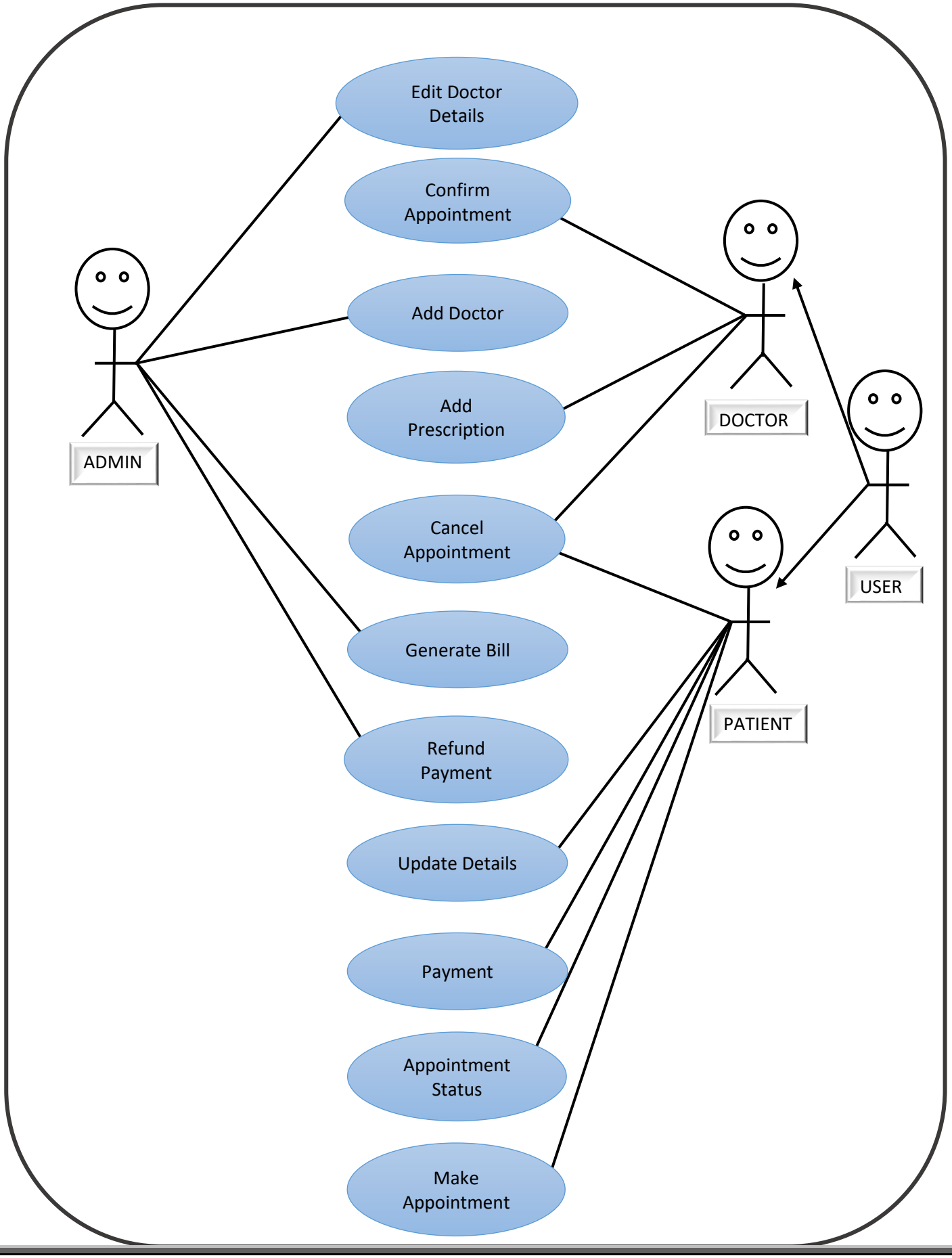
Client End

- **OS** – Windows 7 and above - Very user friendly and common OS
- **MySQL server** - Database connectivity

2.2 Product functions

- Provide access to registered users only
- Registration of new patients.
- Enable patient to view their record.
- Generate appointment date and timing.
- Confirmation by doctor.
- Patient can do payment.
- Modification is schedule by patient.
- Admin can access to patient's record.
- Admin verify payment and generate Bill/Receipt.
- Admin can view monthly/yearly records.

2.3 Use Case Diagram



2.4 Use Case Description

1. Patient

* **Registration**

Description – The new patient can register themselves and their details like name, age, gender, blood group etc. The patient entry will be made in the HMS database.

Pre-Condition – The patient must be a new patient, if necessary, fields left by user, then prompt user to fill the necessary fields.

Main Flow of Events

1. Patient selects sing up in login module.
2. A registration form gets displayed.
3. Patient fills the required details.

Post Conditions – Patient record is added to HMS database

* **Updation**

Description – The patient should be enabled to update his/her details and the changes should reflect in HMS database.

Pre-Condition – The patient must be a registered patient; The patient cannot update details after treatment starts.

Main Flow of Events

1. Patient logs in to the system.
2. Patient view his record
3. Patient selects update details.
4. Now patient may change the necessary fields.
5. Pop of update details.

Post Condition – The record of patient is updated in HMS database.

* **Appointment**

Description – It shows users a list of available doctors, timings, dates and enables patients to select the most suitable appointment date and doctor. The patient may also the cancel the appointment.

Pre-Condition – The patient must be a registered patient; patient can fix only one appointment for ka particular department.

Main Flow of Event

1. Patient first logs in to system.
2. View his/her record.
3. Create a new appointment or cancel the appointment.

Post Condition - patient details are displayed and a new appointment is fix or a existing appointment is cancelled. The HMS database is updated.

* **Payment**

Description – It enables user to pay the consultant fee of doctor online.

Pre-Condition - The patient must be a registered patient, If Patient don't want to pay online, he/she can pay by cash also.

Main Flow of Event

1. Patient first logs in to system.
2. View his/her record.
3. Appointment confirmed by the Doctor then go for Payment.

Post Conditions – A Receipt will be displayed. The HMS database is updated

2. Doctor

Description – The doctor view patient record/update and add description of the treatment given to patient.

Pre-Condition – The doctor must be a registered doctor, system does not allow the doctor to modify the qualification, hospital managed details.

Main Flow of Events

1. Doctor logs in to the system.
2. Doctor may select view patient.
   2.1 Patient record is displayed with treatment history.
3. Doctor adds description of patient treatment.
4. Doctor may select appointment detail.
   4.1 Appointment request is display with schedule.
5. Doctor confirms or cancel appointment.

Post Condition – The patient and doctor's database are updated.

3. Admin

Description – The admin add doctor, update details and verify payment and generate Bill/Receipt for the same.

Main Flow of Events

1. Admin logs in the system.
2. Admin may add doctor new doctor.
   2.1 Admin fills the doctor's details.
3. Admin view Doctor record.
   3.1 Admin enters the doctor id in the system.
   3.2 Doctors details are displayed, Admin can update details.
4. Admin verify the payment submitted by the patient.
   4.1 Generate Bill/Receipt and confirmation message for the same.

Post Condition – The HMS database is updated.

2.5 Use Characteristics

Admin

Admin has full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access to the system.

Key functions:

- Access patient record, doctor Record.
- Add new doctor entry in system database.
- Confirm payment and Generate bill.
- View records. (Total no of patients treated, doctor added/remove, consultant fee).

Patient

Patient can choose the best preferred appointments from the options provided and can also change the appointment schedule or cancel it. After appointment is confirmed by the respective doctor, they can pay their consultant fee online. Patients have access to only their records.

Key functions:

- Make appointment.
- Cancel appointment.
- Update details.
- Payment.
- View Payment History.

Doctor

Doctor can view the patient appointment list and provide the confirmation or make changes in the appointment list if required. Doctors have access to only records of those patients whom they are treating.

Key functions:

- Confirm the appointment.
- Cancel the appointment.
- Modification of appointment list.
- Add prescription.

2.7 Constraints

- System is wirelessly networked with an encryption.
- System is only accessible within the hospital's website only.
- Database is password protected.
- Should use less RAM and processing power.
- Each user should have individual ID and password.
- Only administrator can access the whole system.

2.8 Assumptions and dependencies

- Each user must have a valid user id and password.
- Server must be running for the system function.
- User must log in to the system to access any record.
- Only the administrator can delete records.

3. Specific Requirements

3.1 Performance requirements

- **Response time** – The system will give response within 1 second after checking the patient information and other information.
- **Capacity** – The system must support 1000 people at a time.
- **User interface** – User interface screen will response within 5 seconds.

3.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure. All the administrative and data entry operators have unique logins so system can understand who is login in to system right now no intruders allowed except system administrative nobody cannot change record and valuable data.

3.3 Functional Requirements

S.NO.	MODULE NAME	APPLICABLE ROLE	DESCRIPTION
1	LOGIN	PATIENT DOCTOR ADMIN	**PATIENT**: Can login using unique id and password after this system shall show his/her profile. **DOCTOR**: Can login using unique id and password after this system shall show his/her profile. **ADMIN**: Can login using unique ID and password after this system shall show a profile with links to maintain the website.
2	REGISTRATION	PATIENT	**PATIENT**: Can register by filling all the required details, after this the system will verify the details and check if already registered or not.
3	MAKE APPT.	PATIENT	**PATIENT**: Can select doctor, date, time and make an appointment request after this system shall show a confirmation for appointment request.
4	CANCEL APPT.	PATIENT DOCTOR	**PATIENT**: Can cancel appointment if want to by just one click after this system shall ask for re-schedule or refund of payment. **DOCTOR**: Can cancel appointment if want to by just one click after this system shall send a message to the patient.
5	DOCTOR MODULE	ADMIN	**ADMIN**: Can add a new doctor by filling all the details after this system shall show a confirmation message. Can remove a doctor by just one click after this system shall show confirmation message.
6	PAYMENT	PATIENT	**PATIENT**: Enter payment details and make payment after this system shall show the generated bill by hospital.
7	PATIENT MODULE	PATIENT	**PATIENT**: Can view payment history or can search for a particular bill also after this system shall show a bill or history.

			Can also see or search for a doctor by entering dept. name or doctor ID if known after this system will check for the doctor if found shall show doctor's profile. Can also update details after this system shall ask for re-enter password and after verifying password shall update details.
8	ADD PRESCRIPTION	DOCTOR	**DOCTOR**: Enter patient ID and after this all the treatment details and medicine, remarks and advice for the patient after this system shall show a message for update.

3.4 Non-Functional Requirements

There are a lot of software requirements specifications included in the non-functional requirements of the Hospital Management System, which contains various processes, namely Security, Performance, Maintainability, and Reliability.

Security:

- **Patient Identification:** The system needs the patient to recognize herself or himself using the phone.
- **Logon ID:** Any users who make use of the system need to hold a Logon ID and password.
- **Modifications:** Any modifications like insert, delete, update, etc. for the database can be synchronized quickly and executed only by the ward administrator.
- **Front Desk Staff Rights:** The staff at the front desk can view any data in the Hospital Management system, and add new patients record to the HMS but they don't have any rights to alter any data in it.
- **Administrator rights:** The administrator can view as well as alter any information in the Hospital Management System.

Performance:

- **Response Time:** The system provides acknowledgment in just one second once the 'patient's information is checked.
- **Capacity:** The system needs to support at least 1000 people at once.
- **User-Interface**: The user interface acknowledges within five seconds.
- **Conformity:** The system needs to ensure that the guidelines of the Microsoft accessibilities are followed.

Maintainability:

- **Back-Up:** The system offers efficiency for data backup.
- **Errors:** The system will track every mistake as well as keep a log of it.

Reliability:

- **Availability:** The system is available all the time.

4. Program implementation of SRS

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <conio.h>
#include <time.h>
#include <iomanip>
#include <windows.h>

using namespace std;
int main()

// NOTE: RUN THE PROGRAM IN FULL SCREEN ONLY

{
    system("color 0B");
    char fname[20];
    time_t rawtime;
    struct tm *timeinfo;

    time(&rawtime);
    timeinfo = localtime(&rawtime);

    // printing the welcome note
re:
    cout <<
"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@\n";
    cout << "\t\t\t\t\t@@
_____________________________________________________________________________________ @@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                   WELCOME TO                     |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|               SH BALDEV RAJ MITTAL HOSPITAL
|@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                                                  |@@\n";
    cout << "\t\t\t\t\t@@|                            -Code Written by Priti Kumari
|@@\n";
    cout <<
"\t\t\t\t\t@@|_______________________________________________________________________________
______|@@\n";
    cout <<
"\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n\n\n\n\t\t\t\t
\t";
```

```
    system("pause");
    system("cls");

    int i;
    int login();
    login();
    system("color 31");
    // giving option to the user for their choice
b:
    cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t   SH BALDEV RAJ MITTAL HOSPITAL
\n\n";
    cout << "\n\n\t\t\t\t\t\tPlease,  Choose from the following Options: \n\n";
    cout << "\t\t\t\t\t\t _____________________________________________________________ \n";
    cout << "\t\t\t\t\t\t|                                                 |\n";
    cout << "\t\t\t\t\t\t|          1  >> Add New Patient Record           |\n";
    cout << "\t\t\t\t\t\t|          2  >> Add Diagnosis Information        |\n";
    cout << "\t\t\t\t\t\t|          3  >> Full History of the Patient      |\n";
    cout << "\t\t\t\t\t\t|          4  >> Information About the Hospital   |\n";
    cout << "\t\t\t\t\t\t|          5  >> Exit the Program                 |\n";
    cout << "\t\t\t\t\t\t|_________________________________________________________|\n\n";
a:
    cout << "\t\t\t\t\t\tEnter your choice: ";
    cin >> i;
    if (i > 5 || i < 1)
    {
        cout << "\n\n\t\t\t\t\t\tInvalid Choice\n";
        cout << "\t\t\t\t\tTry again.....\n\n";
        goto a;
    }
    // if inputed choice is other than given choice

    system("cls");

    // displaying the information about the hospital.....option 4
    if (i == 4)
    {
        system("color 02");
        ifstream file;
        file.open("hos.txt");
        if (!file)
        {
            cout << "\nError while opening the file\n";
            goto b;
        }
        else
        {
            cout << "\n\n\n\n\n\n\n\t\t\t\t   .....Information about the
Hospital.....\n\n";
            string line;
            while (file.good())
            {
                getline(file, line);
                cout << line << "\n\t\t";
            }
            cout << "\n\n\t\t";
            system("pause");
            system("cls");
            goto b;
        }
```

```
    }

    // Adding the record of the new patient.....option 3
    if (i == 1)
    {
        system("color F4");
        time_t rawtime;
        struct tm *timeinfo;

        time(&rawtime);
        timeinfo = localtime(&rawtime);
        cout << "\n\n\t\t\t\t\t\t\t\t" << asctime(timeinfo);
        ofstream pat_file;
        char fname[20];
        cout << "\n\n\n\nEnter the patient's file name : ";
        cin.ignore();
        gets(fname);
        pat_file.open(fname);
        if (!fname)
        {
            cout << "\nError while opening the file\n";
            goto b;
        }
        else
        {
            struct patient_info
            {
                char name[20];
                char address[100];
                char contact[10];
                char age[5];
                char sex[8];
                char blood_gp[5];
                char disease_past[50];
                char id[15];
            };

            patient_info ak;
            cout << "\n*********************************************************************\n";
            pat_file << "\n*********************************************************************\n\n"; //
fn1353 st
            cout << "\nName : ";
            pat_file << "Name : ";
            gets(ak.name);
            pat_file << ak.name << "\n";
            cout << "\nAddress : ";
            pat_file << "Address : ";
            gets(ak.address);
            pat_file << ak.address << "\n";
            cout << "\nContact Number : ";
            pat_file << "Contact Number : ";
            gets(ak.contact);
            pat_file << ak.contact << "\n";
            cout << "\nAge : ";
            pat_file << "Age : ";
            gets(ak.age);
            pat_file << ak.age << "\n";
            cout << "\nSex : ";
            pat_file << "Sex : ";
```

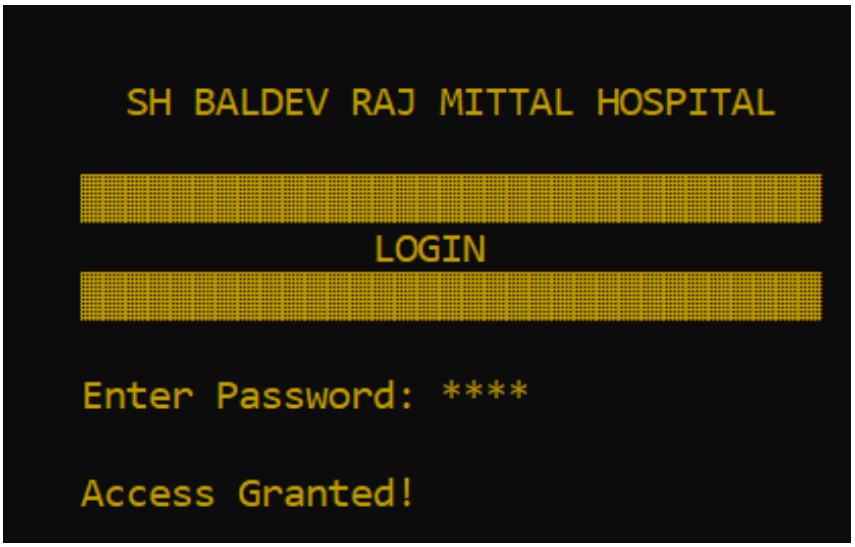
```
            gets(ak.sex);
            pat_file << ak.sex << "\n";
            cout << "\nBlood Group : ";
            pat_file << "Blood Group : ";
            gets(ak.blood_gp);
            pat_file << ak.blood_gp << "\n";
            cout << "\nAny Major disease suffered earlier : ";
            pat_file << "Any Major disease suffered earlier : ";
            gets(ak.disease_past);
            pat_file << ak.disease_past << "\n";
            int i;
            srand(time(NULL));              //Function for randomly generate the patient ID
            i = (rand() % 99999999 + 1);

            pat_file << "Patient ID : ";
            pat_file << i << "\n";
            cout << "\n*********************************************************************************\n";
            pat_file << "\n*********************************************************************************\n\n";
            cout << "\nInformation Saved Successfully\n";
        }
        system("pause");
        system("cls");
        goto b;
    }

    // Appending diagnosis information of patient datewise.....option 2
    if (i == 2)
    {
        system("color C");
        fstream pat_file;
        cout << "\n\nEnter the patient's file name to be opened : ";
        cin.ignore();
        gets(fname);
        system("cls");
        pat_file.open(fname, ios::in);
        if (!pat_file)
        {
            cout << "\nError while opening the file\n";
            goto b;
        }
        else
        {
            cout << "\n\n\n\n\t\t\t\t..... Information about " << fname << "
.....\n\n\n\n";
            string info;
            while (pat_file.good())
            {
                getline(pat_file, info);
                cout << info << "\n";
            }
            cout << "\n";
            pat_file.close();
            pat_file.open(fname, ios::out | ios::app);
            cout << "\n";
            cout << "Adding more information in patient's file.....on : " << asctime(timeinfo);
            pat_file << "Description of " << asctime(timeinfo) << "\n";
            struct app
            {
                char symptom[500];
```

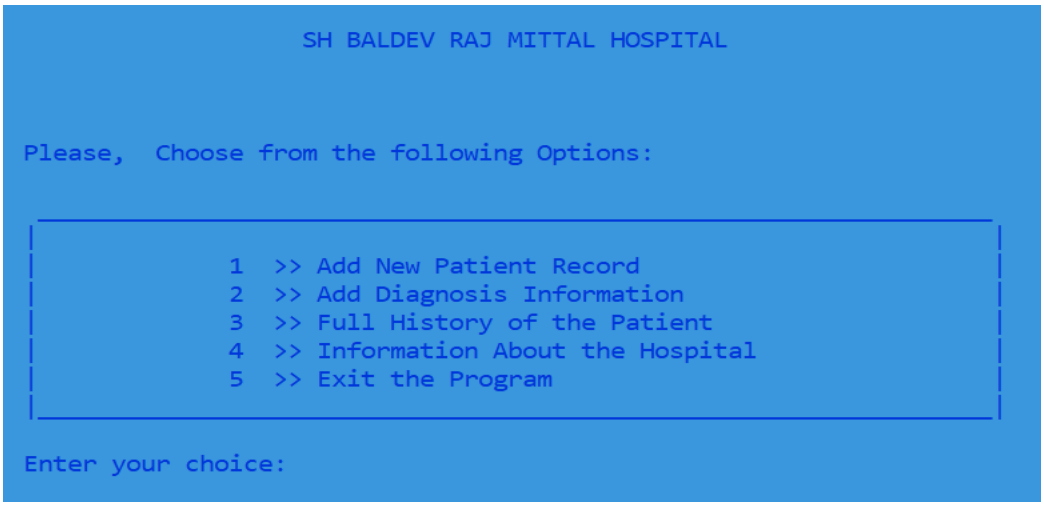
```
            char diagnosis[500];
            char medicine[500];
            char addmission[30];
            char ward[15];
        };
        app add;
        cout << "\nSymptoms : ";
        pat_file << "Symptoms : ";
        gets(add.symptom);
        pat_file << add.symptom << "\n";
        cout << "\nDiagnosis : ";
        pat_file << "Diagnosis : ";
        gets(add.diagnosis);
        pat_file << add.diagnosis << "\n";
        cout << "\nMedicines : ";
        pat_file << "Medicines : ";
        gets(add.medicine);
        pat_file << add.medicine << "\n";
        cout << "\nAddmission Required? : ";
        pat_file << "Addmission Required? : ";
        gets(add.addmission);
        pat_file << add.addmission << "\n";
        cout << "\nType of ward : ";
        pat_file << "Type of ward : ";
        gets(add.ward);
        pat_file << add.ward << "\n";
        pat_file << "\n*********************************************************************************\n";
        cout << "\n\n"
             << add.ward << " ward is alloted Successfully\n";
        pat_file.close();
        cout << "\n\n";
        system("pause");
        system("cls");
        goto b;
    }
}

// For displaying the full medical history of patient in that hospital.....option 3
if (i == 3)
{
    system("color 6");
    fstream pat_file;
    cout << "\n\nEnter the patient's file name to be opened : ";
    cin.ignore();
    gets(fname);
    system("cls");
    pat_file.open(fname, ios::in);
    if (!pat_file)
    {
        cout << "\nError while opening the file\n";
        goto b;
    }
    else
    {
        cout << "\n\n\n\n\t\t\t\t..... Full Medical History of " << fname << "
.....\n\n\n\n";
        string info;
        while (pat_file.good())
        {
```

```
            getline(pat_file, info);
            cout << info << "\n";
        }
        cout << "\n";
    }
    system("pause");
    system("cls");
    goto b;
}

// Exiting Through the system with a Thank You note.....option 5
if (i == 5)
{
    system("cls");
    system("color DF");

    cout <<
"\n\n\t\t\t\t\t**********************************************************************************" << endl;
    cout << "\t\t\t\t\t*\t\t\t\t\t\t\t\t       *" << endl;
    cout << "\t\t\t\t\t*\t\t\t\t\t\t\t\t       *" << endl;
    cout << "\t\t\t\t\t*   _____   _   _     _     _   _   _  __   __   __   ___    _   _    *" << endl;
    cout << "\t\t\t\t\t*  |_   _|| | | |   / \\   | \\ | | | |/ /   \\ \\ \\ / /  / _ \\  | | | |  *" << endl;
    cout << "\t\t\t\t\t*    | |  | |_| |  / _ \\  |  \\| | | ' /     \\ V / | | | | | | |  *" << endl;
    cout << "\t\t\t\t\t*    | |  |  _  | / ___ \\ | |\\  | |  \\      | |  | |_| | |_| |  *" << endl;
    cout << "\t\t\t\t\t*    |_|  |_| |_|/_/   \\_\\|_| \\_| |_|\\_\\     |_|   \\___/   \\___/   *" << endl;
    cout << "\t\t\t\t\t*\t\t\t\t\t\t\t\t       *" << endl;
    cout << "\t\t\t\t\t*\t\t\t\t\t\t\t\t       *" << endl;
    cout << "\t\t\t\t\t*\t\t\t      FOR CHOOSING                              *" << endl;
    cout << "\t\t\t\t\t*\t\t\t SH BALDEV RAJ MITTAL HOSPITAL                   *" << endl;
    cout <<
"\t\t\t\t\t**********************************************************************************\n\n";
}

    cout << "\n";
}

int login()
{
    system("color 6");
    string pass = "";
    char ch;
    cout << "\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t  SH BALDEV RAJ MITTAL HOSPITAL
\n\n";
    cout <<
"\t\t\t\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
B2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";
    cout << "\n\t\t\t\t\t\t\t\t     LOGIN \n";
    cout <<
"\t\t\t\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
B2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\n\n";
    cout << "\t\t\t\t\t\t\tEnter Password: ";
    ch = _getch();
    while (ch != 13)
    { // character 13 is enter
        pass.push_back(ch);
        cout << '*';
        ch = _getch();
    }
    if (pass == "pass")
```

```
    {
        cout << "\n\n\t\t\t\t\t\t\t\tAccess Granted! \n";
        system("PAUSE");
        system("CLS");
    }
    else
    {
        cout << "\n\n\t\t\t\t\t\t\t\tAccess Denind.....\n\t\t\t\t\t\t\t\tPlease Try Again\n\n";
        system("PAUSE");
        system("CLS");
        login();
    }
}
```

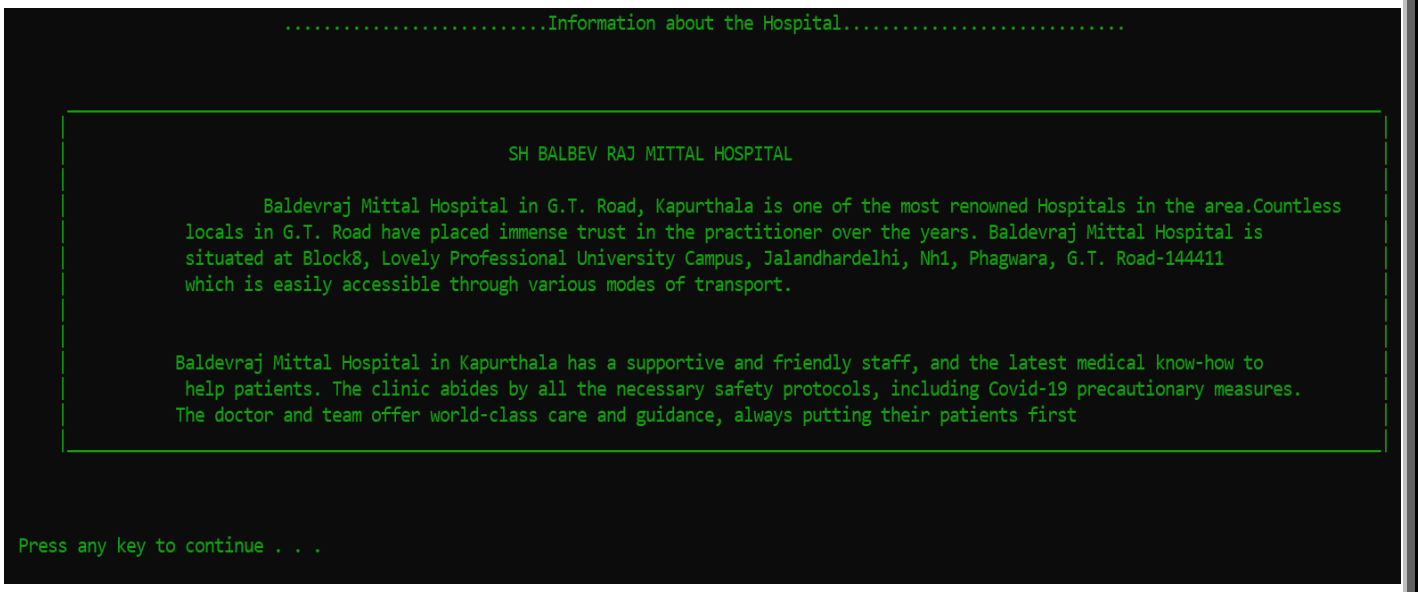
5. OUTPUT:-


SH BALDEV RAJ MITTAL HOSPITAL
LOGIN
Enter Password: ****
Access Granted!


SH BALDEV RAJ MITTAL HOSPITAL
Please, Choose from the following Options:
1 >> Add New Patient Record
2 >> Add Diagnosis Information
3 >> Full History of the Patient
4 >> Information About the Hospital
5 >> Exit the Program
Enter your choice:

```
                                                        Tue Dec 06 19:25:53 2022



Enter the patient's file name : priti singh

*****************************************************************

Name : priti singh

Address : patna

Contact Number : 2356789066

Age : 23

Sex : Female

Blood Group : o+

Any Major disease suffered earlier : No

*****************************************************************

Information Saved Successfully
Press any key to continue . . .
```

```
                          ..................................... Full Medical History of priti singh .........



*****************************************************************

Name : priti singh
Address : patna
Contact Number : 2356789066
Age : 23
Sex : Female
Blood Group : o+
Any Major disease suffered earlier : No
Patient ID : 17335

*****************************************************************



Press any key to continue . . .
```


.....Information about the Hospital.....
SH BALBEV RAJ MITTAL HOSPITAL
Baldevraj Mittal Hospital in G.T. Road, Kapurthala is one of the most renowned Hospitals in the area.Countless locals in G.T. Road have placed immense trust in the practitioner over the years. Baldevraj Mittal Hospital is situated at Block8, Lovely Professional University Campus, Jalandhardelhi, Nh1, Phagwara, G.T. Road-144411 which is easily accessible through various modes of transport.
Baldevraj Mittal Hospital in Kapurthala has a supportive and friendly staff, and the latest medical know-how to help patients. The clinic abides by all the necessary safety protocols, including Covid-19 precautionary measures. The doctor and team offer world-class care and guidance, always putting their patients first
Press any key to continue . . .


THANK YOU
FOR CHOOSING
SH BALDEV RAJ MITTAL HOSPITAL