

BAHRIA UNIVERSITY KARACHI CAMPUS
DEPARTMENT OF COMPUTER SCIENCE



Bahria University
Discovering Knowledge

CLINICAL MANAGEMENT SYSTEM
COMPUTER PROGRAMMING LAB
CSL 113

Alishba Siraj 02-134222-005

Hamza Altaf 02-134222-006

Nahin Fatima 02-134222-102

Report should cover these heading:

1. Acknowledge
2. Abstract
3. List of contexts
4. List of figures
5. Introduction
6. Problem and solution
7. Workflow
8. Overview of project
9. Output (screen shots with details)
10. Conclusion
11. References

GUIDELINES FOR PROJECT REPORT

1. The project report should be hard bound
2. The front cover of the report should be in printed form along-with the university logo (new).
3. The font used should be Times New Roman, 12 points plain, with one-and-a-half (1.5) spacing between two consecutive lines
4. For sub-headings, increase one point and for main headings increase one more point.
5. All the pages should be properly numbered.

ACKNOWLEDGE:

We take this opportunity to acknowledge everyone who have helped us in every stage of this project.

Firstly, I am indebtedly grateful to our computer programming lab teacher Miss. Tahira for guiding us and making us understand C++ programming language. We are also thankful to our Theory teacher Miss. Azeema for her valuable instructions. Without their support and teaching this project would not have been completed.

ABSTRACT:

A clinic is a health care facility that is primarily devoted to the care of all patients. Clinic can be privately operated or publicly manage and funded and typically care their primary health care needs of population in local communities, in contracts to larger hospital which gives specialized treatment and admit in patient box our night stays.

LIST OF CONTEXTS:

- Store the patient`s information.
- Generate an appointment.
- Cancel an appointment.
- Generate the bill.

LIST OF FIGURES:

- Introduction
- Problem and solution
- Workflow
- Overview of project
- Output (screen shots with details)
- Conclusion
- References

INTRODUCTION:

This Clinic Management System can enhance the services and the workflow of all activity that happens in hospitals where it helps in reducing the workload of medical staff, the number of manpower needed, and it also make hospitals management become more manageable and easier to control. Currently, there is a huge number of private and public hospitals that keep their patients records in books and store them manually. it's a very formal way of storing data and records but the demerit of storing data in this manner is poor handling of the books may lead to lose of data, the books are not that much secure and are easily perishable furthermore books have no back up once lost cannot be retrieved. Applying this system in such institutions will help manage these records and preserve the information for longer periods of time and make its access far much simpler and easier, it also ensures accuracy and openness of the patient's details / records.

PROBLEM AND SOLUTION:

- **Large space usage.** Use of paper to record data requires large space for storage over time. This can be seen in many places where analogue systems are in use.
So Clinical Management system fixes the analogue systems problem by storing data in the form of files.
- **Inefficient back up methods.** Old systems especially that which are analogue do not provide effective ways to back up data and as a result data can be permanently lost in the event data is damaged or misplaced. This may lead to the collapse of the institution. Analogue systems do not allow efficient supervision of its users and thus employees are likely to commit fraud undetected and get away with it. Now the systems are backed up and encrypted to avoid loss of data and data theft.

WORKFLOW:

Patient registration: Patients register themselves by providing their personal and medical information.

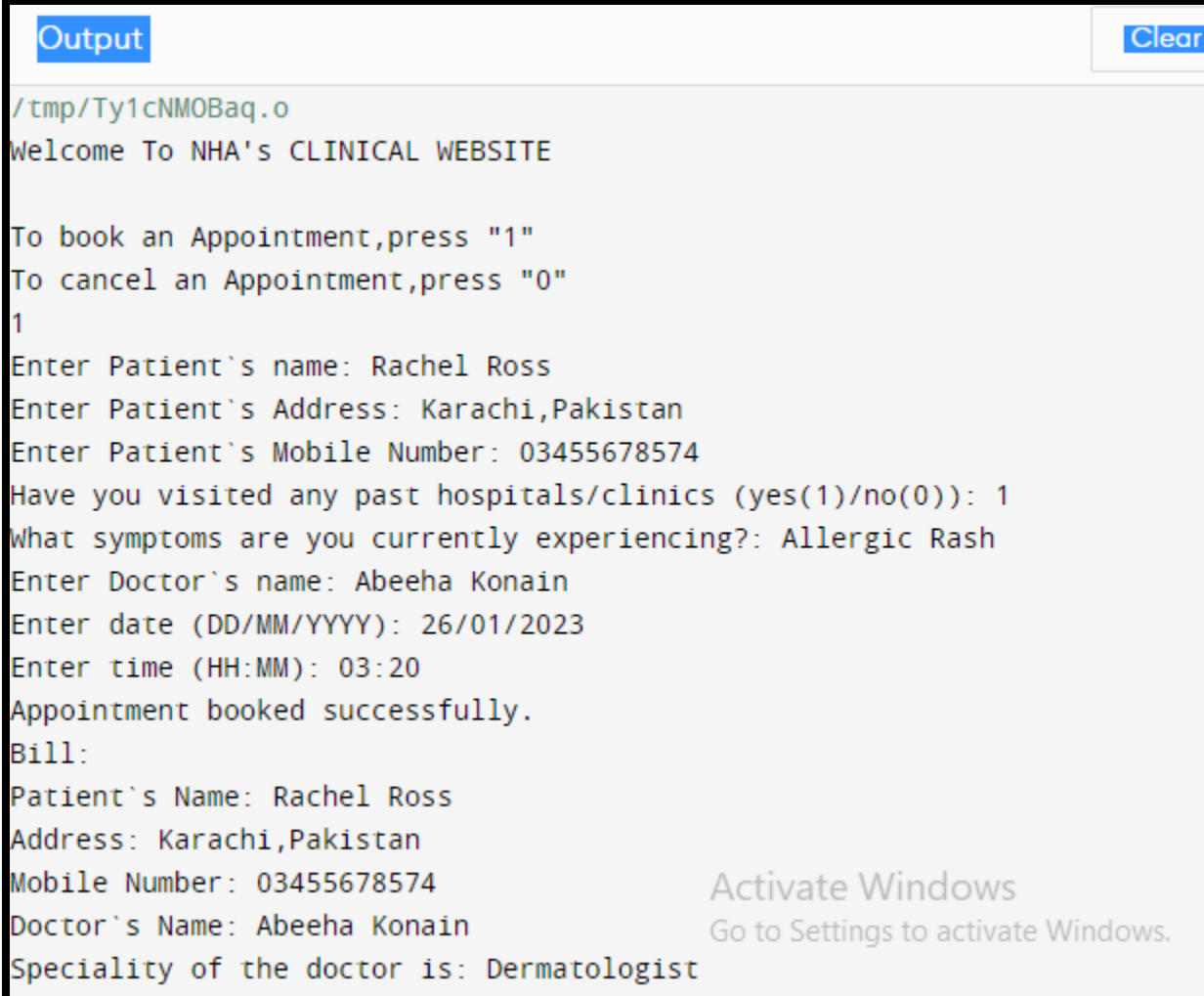
Appointment scheduling: Patients can schedule appointments with doctors and view their upcoming appointments.

Billing and payment: Patients make payment for their consultation and treatment, and the system generates a bill and maintains a record of the transaction

Consultation and treatment: Patients visit the clinic for consultation and treatment, and doctors update the patient records accordingly.

OVERVIEW OF PROJECT:

A Clinic Management System is a computer software designed to manage and organize the day-to-day operations of a clinic. It typically includes a range of features that help automate and streamline various processes, such as patient registration, appointment scheduling, billing, prescription management, and data analysis. The system is usually designed to be user-friendly and easy to navigate, with secure access controls to protect sensitive patient information.



```
Output
/tmp/Ty1cNMOBaq.o
Welcome To NHA's CLINICAL WEBSITE

To book an Appointment,press "1"
To cancel an Appointment,press "0"
1
Enter Patient's name: Rachel Ross
Enter Patient's Address: Karachi,Pakistan
Enter Patient's Mobile Number: 03455678574
Have you visited any past hospitals/clinics (yes(1)/no(0)): 1
What symptoms are you currently experiencing?: Allergic Rash
Enter Doctor's name: Abeeha Konain
Enter date (DD/MM/YYYY): 26/01/2023
Enter time (HH:MM): 03:20
Appointment booked successfully.
Bill:
Patient's Name: Rachel Ross
Address: Karachi,Pakistan
Mobile Number: 03455678574
Doctor's Name: Abeeha Konain
Speciality of the doctor is: Dermatologist

Activate Windows
Go to Settings to activate Windows.
```

```
Appointment Time: 03:20
Appointment Date: 26/01/2023
Price: $150
    Get well soon =D
Thank you!
```

CONCLUSION:

The main goal of this system is to improve the efficiency and effectiveness of clinic operations, as well as providing better services to patients. By automating repetitive tasks, the system frees up clinic staff to focus on more important tasks, such as providing medical care. Additionally, by providing real-time access to patient information, the system enables doctors to make more informed decisions and provide better care.

REFERENCES:

- [https://ir.unimas.my/id/eprint/3354/1/Clinical%20management%20system%20\(CMS\).pdf](https://ir.unimas.my/id/eprint/3354/1/Clinical%20management%20system%20(CMS).pdf)
- https://www.academia.edu/13414341/CLINICAL_MANAGEMENT_SYSTEM_PROJECT_DOCUMENTATION
- <https://stackoverflow.com>
- <https://github.com>
- <https://stackoverflow.com/questions/9965710/how-to-change-text-and-background-color>
- <https://www.geeksforgeeks.org>

Source Code:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

const int MAX_APPOINTMENTS = 100; // maximum number of appointments that can
be stored

// appointment structure
struct Appointment
{
    string patientName;
    string patientaddress;
    string mobileNumber;
    bool pastHospitalVisit;
    string symptoms;
    string doctorName;
    string date;
    string time;
};

// doctor structure
struct Doctor
{
    string doctorName;
    string specialty;
};

// array to store doctor information
Doctor doctorinfo[4];
string doctorinformation(string x)
{
    string ans;
    doctorinfo[0].doctorName = "Alishba Siraj";
    doctorinfo[1].doctorName = "Hamza Altaf";
    doctorinfo[2].doctorName = "Nahin Fatima";
    doctorinfo[3].doctorName = "Abeeha Konain";
    doctorinfo[0].specialty = "Ophthalmologists";
    doctorinfo[1].specialty = "Orthopedic";
    doctorinfo[2].specialty = "Pediatrician";
    doctorinfo[3].specialty = "Dermatologist";
    for (int i = 0; i < 4; i++)
    {
        if (x == doctorinfo[i].doctorName)
        {
            ans = doctorinfo[i].specialty;
        }
    }
}
```

```

    }
    }
    return ans;
}

// array to store appointments
Appointment appointments[MAX_APPOINTMENTS];
int appointmentCount = 0; // current number of appointments

// function to book an appointment
void bookAppointment()
{
    if (appointmentCount >= MAX_APPOINTMENTS)
    {
        cout << "Sorry, no more appointments can be booked." << endl;
        return;
    }
    string answer;
    cout << "Enter Patient's name: ";
    getline(cin, appointments[appointmentCount].patientName);
    getline(cin, appointments[appointmentCount].patientName);
    cout << "Enter Patient's Address: ";
    getline(cin, appointments[appointmentCount].patientaddress);
    cout << "Enter Patient's Mobile Number: ";
    cin >> appointments[appointmentCount].mobileNumber;
    cout << "Have you visited any past hospitals/clinics (yes(1)/no(0))?: ";
    cin >> appointments[appointmentCount].pastHospitalVisit;
    answer = appointments[appointmentCount].pastHospitalVisit;
    if (answer == "yes")
    {
        appointments[appointmentCount].pastHospitalVisit = true;
    }
    else
    {
        appointments[appointmentCount].pastHospitalVisit = false;
    }
    cout << "What symptoms are you currently experiencing?: ";
    getline(cin, appointments[appointmentCount].symptoms);
    getline(cin, appointments[appointmentCount].symptoms);
    cout << "Enter Doctor's name: ";
    getline(cin, appointments[appointmentCount].doctorName);
    cout << "Enter date (DD/MM/YYYY): ";
    getline(cin, appointments[appointmentCount].date);
    cout << "Enter time (HH:MM): ";
    getline(cin, appointments[appointmentCount].time);

    appointmentCount++;

    cout << "Appointment booked successfully." << endl;
}

// function to cancel an appointment
void cancelAppointment()

```



```

{
    string patientName;
    cout << "Enter Patient's name: ";
    getline(cin, patientName);
    getline(cin, patientName);
    // search for the appointment
    int index = -1;
    for (int i = 0; i < appointmentCount; i++)
    {
        if (appointments[i].patientName == patientName)
        {
            index = i;
            break;
        }
    }

    if (index == -1)
    {
        cout << "Appointment not found." << endl;
    }
    else
    {
        // remove the appointment by shifting the array
        for (int i = index; i < appointmentCount - 1; i++)
        {
            appointments[i] = appointments[i + 1];
        }
        appointmentCount--;

        cout << "Appointment cancelled successfully." << endl;
    }
}

// function to save appointments to a file
void saveAppointments()
{
    string s;
    ofstream file("appointments.txt");
    for (int i = 0; i < appointmentCount; i++)
    {
        s = doctorinformation(appointments[appointmentCount].doctorName);
        file << "Patient's Name: " << appointments[i].patientName << "
        Doctor's Name: " << appointments[i].doctorName << "    Appointment's Date: "
        << appointments[i].date << "    Appointment's Time:" << appointments[i].time
        << "    Doctor's Specialty: " << s << endl;
    }
    file.close();
}

// function to load appointments from a file
void loadAppointments()
{
    ifstream file("appointments.txt");

```

```

        appointmentCount = 0;
        while (file >> appointments[appointmentCount].patientName) {
            file >> appointments[appointmentCount].doctorName >>
appointments[appointmentCount].date >> appointments[appointmentCount].time;
            appointmentCount++;
        }
        file.close();
    }
    // Generate bill
    void Generatebill(Appointment appointments[MAX_APPOINTMENTS])
    {
        string specialty =
doctorinformation(appointments[appointmentCount].doctorName);
        cout << "Bill: " << endl;
        cout << "Patient's Name: " << appointments[appointmentCount].patientName
<< endl;
        cout << "Address: " << appointments[appointmentCount].patientaddress <<
endl;
        cout << "Mobile Number: " << appointments[appointmentCount].mobileNumber
<< endl;
        cout << "Doctor's Name: " + appointments[appointmentCount].doctorName <<
"Speciality of the doctor is: " << specialty << endl;
        cout << "Appointment Time: " + appointments[appointmentCount].time <<
endl;
        cout << "Appointment Date: " + appointments[appointmentCount].date <<
endl;
        cout << "Price: $150 \n" << "\t Get well soon =D \n";
    }

    int main() {
        int option;
        while (true)
        {
            cout << "
CLINICAL WEBSITE                               Welcome To NHA's
            " << endl << endl;
            cout << "To book an Appointment,press \"1\"" << endl;
            cout << "To cancel an Appointment,press \"0\"" << endl;
            cin >> option;
            switch (option)
            {
                case 1:
                    bookAppointment();
                    saveAppointments();
                    loadAppointments();
                    Generatebill(appointments);
                    break;
                case 0:
                    cancelAppointment();
                    loadAppointments();
                    break;
                default:
                    cout << "Invalid Input" << endl;
            }
        }
    }
}

```

```
    }  
    cout << "Thank you!";  
    return 0;  
  }  
}
```

Flowchart:

