

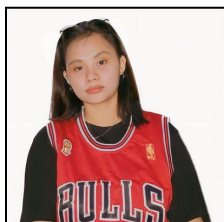


COMPUTER PROGRAMMING II

(ITC 123L)



Asadon, Liane
CEIT - 37 - 202P



Barcebas, Mesha
CEIT - 37 - 202P



Collado, Jonalyn
CEIT - 37 - 202P



Eulalio, Emmanuel
CEIT - 37 - 202P



Garong, Maeshell
CEIT - 37 - 202P



Grayda, Ian Carl
CEIT - 37 - 202P



Jabinales, Nathaniel
CEIT - 37 - 202P



Sese, Angelo
CEIT - 37 - 202P

TABLE OF CONTENTS

I. INTRODUCTION.....	2
II. DESCRIPTION OF THE STUDY.....	2
III. OBJECTIVES.....	3
IV. SIGNIFICANCE OF THE STUDY.....	3
V. SCOPE AND DELIMITATION.....	4
VI. SCREEN OUTPUT.....	5
VII. SOURCE CODE.....	7
VIII. CONCLUSION.....	7



HOSPITAL MANAGEMENT SYSTEM

I. INTRODUCTION

The advocacy of our research is to make it more convenient for patients to records everything through our system. Enhances information integrity, reduces transcription errors, and reduces duplication of information entries. Hospital management system is a system to show all the details and for faster transactions to the patients. The system "Hope Medical Center " is combined from the word "Hope " and "Center " or in the other words people who's hoping to be cured. Hope Medical Center System are all terms for an establishment that primarily serves data and other features. Hope Medical Center or Hospital management system focuses on the patients private information instead of writing in paper and fill all the necessary info, they just need to fill out through their electronic devices wich is more convenient.

II. DESCRIPTION OF THE STUDY

As you can see throughout the project we have been creating in the past few weeks, we have managed to create a system that will help the HMC (Hope Medical Center) by putting doctors and employees at ease when finding documents about the transactions and records of the patients that are confined in the hospital. We researchers provide special features for the users of the hospital to make it more efficient and suitable to use, which may help them manage the facilities and hospital more conveniently. Furthermore, we are still developing and improving this system, adding more features that suit every user and will satisfy them the most.

Features include the following:



COLLEGE OF ENGINEERING, ARCHITECTURE AND TECHNOLOGY

- **Add new patient record:**

-This section is where you can register the patient's new information, such as name, address, and contact information.

- **Add diagnosis information:**

-In this section, you can save the illness that you diagnose for the patient, and there is information needed such as symptoms, diagnosis, and medicine.

- **Full history of the patient:**

-In this section, you can see all of the full information that has been registered in the patient record and the diagnosis information.

III. OBJECTIVES

The main objective of this system is to track down the full history of the patient by sorting the patient out, it also helps the doctors and nurses to find your past record if you have one, if you don't this program will be used to save and make a new record for you (such as: the patient's diagnosis, patient's information and history). By using this system we can ease the work of employees a little bit, by organizing the records of the patients, it is also easy to pass on the record information to nurses and to doctors so it will not be a hassle to them.

IV. SIGNIFICANCE OF THE STUDY

The study of hospital management system is significant for several reasons

1. **Efficient patient care:** A hospital management system helps hospitals to keep accurate and up-to-date records of patients, their medical history, and treatment plans. This



enables doctors and other healthcare providers to provide more efficient healthcare services to patients.

2. Improved communication: Hospital management systems also facilitate better communication among healthcare providers and departments. With the system, doctors, nurses, and administrative staff can share information and track patients' progress from one department to the next.

3. Cost-effective: Implementing a hospital management system can reduce the cost of managing hospitals. The system can help healthcare providers to reduce paperwork, minimize redundancy, and optimize time and resources.

4. Better patient outcomes: The use of a hospital management system can lead to better patient outcomes as it helps to identify potential health problems in a timely manner. With the system, healthcare providers can get real-time alerts on patient progress, medication use and potential side effects. This enables them to take necessary action to prevent adverse events.

The study of hospital management system is essential for improving the overall quality of healthcare delivered in hospitals.

V. SCOPE AND DELIMITATION

The scope and delimitation of this study is to design and develop a hospital management system that can efficiently manage the records and transactions of patients, doctors, nurses, and other staff in a hospital. The system will provide a user-friendly interface for the users to access and update the information in the database. The system will also generate reports and statistics for the management to monitor and evaluate the performance of the hospital.



The delimitation of this study is that it will only focus on the core functionalities of a hospital management system, such as patient registration, Diagnosis information, and medical records. The system will not include other features that are not directly related to the hospital operations, such as online booking, telemedicine, insurance claims, or customer feedback. The system will also not cover the security and privacy issues that may arise from storing and processing sensitive data in the cloud.

VI. SCREEN OUTPUT

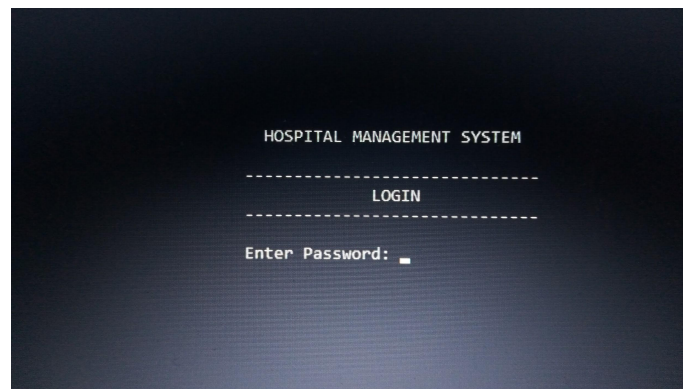


FIGURE 1. Ask to enter a password to access the patient documents or enter patient information.

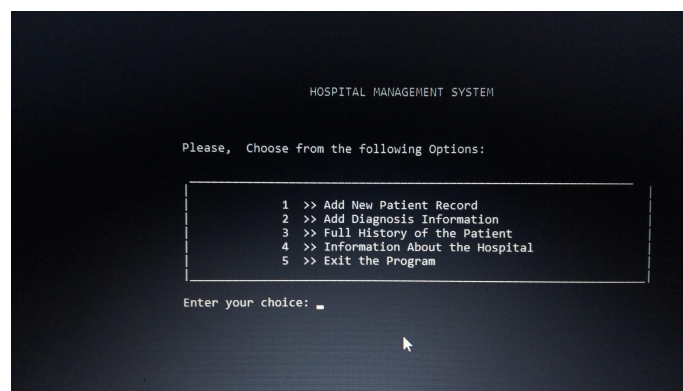


FIGURE 2. Ask to choose from the following options

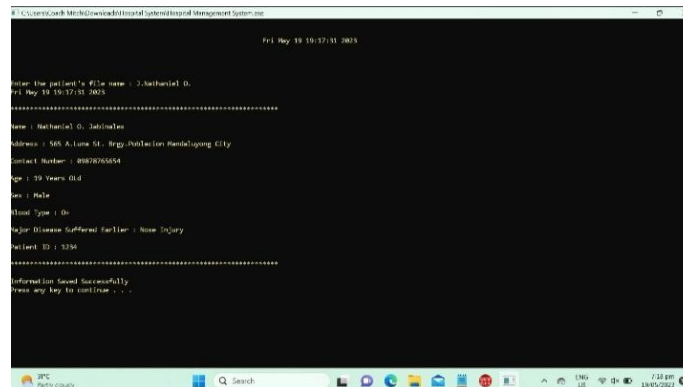


Figure 3. Ask the patient personal details

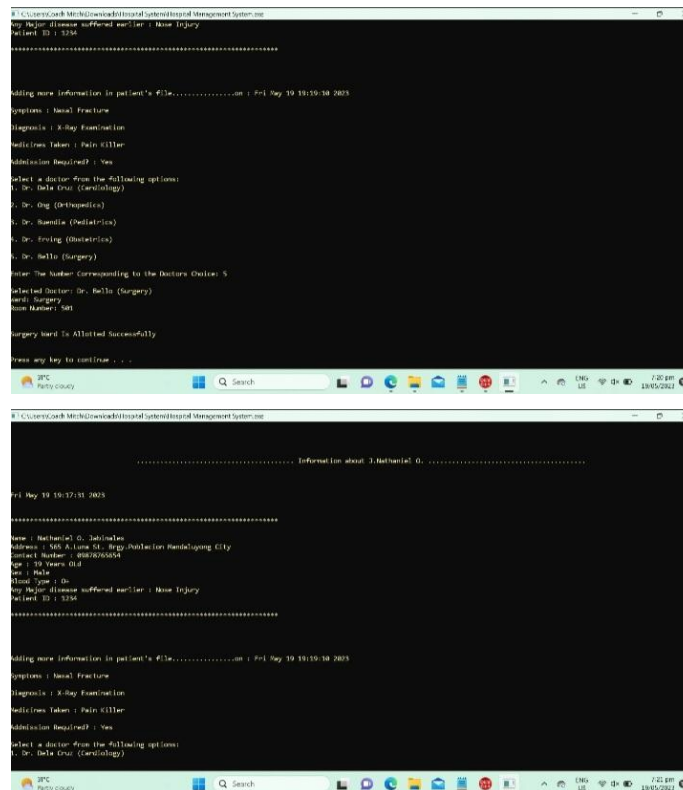


Figure 4. Ask for additional findings to the patients and will display the information of the patient.

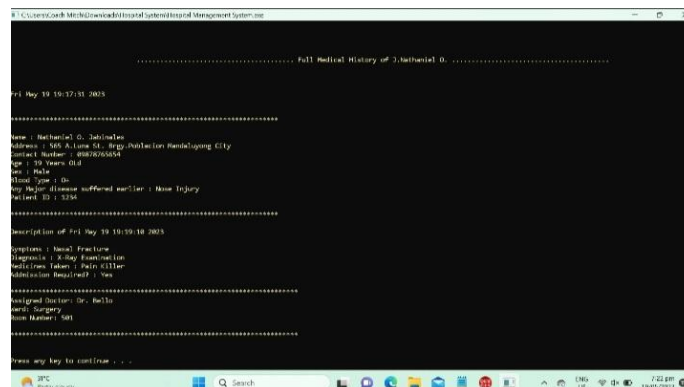


Figure 5. Will display the full medical history of the patient.

VII. SOURCE CODE

```
#include<iostream>
#include<fstream>
#include<cstdlib>
#include<conio.h>
#include<time.h>
#include<iomanip>
#include <windows.h>
#include <stdio.h>
using namespace std;

int main() {

    system("COLOR 0e");
    system("cls");
    printf("\n[?25l]");

    SetConsoleCP(437);
    SetConsoleOutputCP(437);
    int barl = 222;

    cout<<"\n\n\n\n\n\n\n\t\t\t\t\t\n";
    cout<<"\t\t\t\t\t
```

[illegible]

```

";
Beep(1000, 1000);
    Beep(500, 1000);
    Beep(400, 1000);
    Beep(300, 1000);
    Beep(200, 1000);
    Beep(100, 1000);
    Beep(50, 1000);

```

```
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\t\t\t\t\t\t+++++  
+++++\n";  
cout<<"\t\t\t\t\t\t++
```

[illegible]

[illegible]



```
//---Display Information About Hospital---
if(i==4)
{
    ifstream file;
    file.open("hos");
    if(!file)
    {
        cout<<"\nError while opening the file\n";goto b;
    }
    else
    {
        cout<<"\n\n\n\n\n\n\n\t\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 Record For New Patient---
if(i==1)
{
    time_t rawtime;
    struct tm * timeinfo;

    time ( &rawtime );
    timeinfo = localtime ( &rawtime );
    cout<<"\n\n\t\t\t\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);
}
```



COLLEGE OF ENGINEERING, ARCHITECTURE AND TECHNOLOGY

```

        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];
            };
            cout<<asctime (timeinfo);pat_file<<asctime (timeinfo)<<"\n";
            patient_info ak;

            cout<<"\n*****\n";
            pat_file<<"\n*****\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 Type : ";pat_file<<"Blood Type : ";
            gets(ak.blood_gp);pat_file<<ak.blood_gp<<"\n";
            cout<<"\nMajor Disease Suffered Earlier : ";pat_file<<"Any Major disease suffered
            earlier : ";gets(ak.disease_past);pat_file<<ak.disease_past<<"\n";
            cout<<"\nPatient ID : ";pat_file<<"Patient ID : ";gets(ak.id);pat_file<<ak.id<<"\n";

            cout<<"\n*****\n";
            pat_file<<"\n*****\n";
            *"\n\n";
            cout<<"\nInformation Saved Successfully\n";
        }
        system("pause");
        system("cls");

```



```
goto b;

}

struct Doctor {
    string name;
    string ward;
    int roomNumber;
};

//---ARRAY FOR DOCTORS INFORMATION---
Doctor doctors[5];

//---ASSIGNING INFORMATION TO DOCTORS---
doctors[0] = {"Dr. Dela Cruz", "Cardiology", 101};
doctors[1] = {"Dr. Ong", "Orthopedics", 201};
doctors[2] = {"Dr. Buendia", "Pediatrics", 301};
doctors[3] = {"Dr. Erving", "Obstetrics", 401};
doctors[4] = {"Dr. Bello", "Surgery", 501};

//---ADD DIAGNOSIS INFORMATION FOR PATIENT---
if(i==2)
{
    time_t rawtime;
    struct tm * anothertime;

    time ( &rawtime );
    anothertime = localtime ( &rawtime );
    cout<<"\n\n\t\t\t\t\t\t\t\t\t\t" << asctime (anothertime);

    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\t\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 (anothertime);pat_file<<"Description of "<<asctime (anothertime)<<"\n";
        struct app
        {
            char symptom[500];
            char diagnosis[500];
            char medicine[500];
            char admission[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 Taken : "; pat_file<<"Medicines Taken : ";gets(add.medicine);
        pat_file<<add.medicine<<"\n";
        cout<<"\nAdmission Required? : "; pat_file<<"Admission Required? :
";gets(add.admission);
        pat_file<<add.admission<<"\n";pat_file<<"\n*****
*****\n";
        cout<<"\n";

        cout << "Select a doctor from the following options:\n";
        for (int j = 0; j < 5; j++) {
            cout << j + 1 << ". " << doctors[j].name << " (" << doctors[j].ward << ")\n";
            cout<<"\n";
        }
    }

```



```
int choice;
cout << "Enter The Number Corresponding to the Doctors Choice: ";
cin >> choice;

if (choice >= 1 && choice <= 5) {
    Doctor assignedDoctor = doctors[choice - 1];
    cout << "\nSelected Doctor: " << assignedDoctor.name << " (" << assignedDoctor.ward
<< ") \n";
    cout << "Ward: " << assignedDoctor.ward << "\n";
    cout << "Room Number: " << assignedDoctor.roomNumber << "\n";

    // Update the patient's file with the assigned doctor and ward information
    pat_file << "Assigned Doctor: " << assignedDoctor.name << "\n";
    pat_file << "Ward: " << assignedDoctor.ward << "\n";
    pat_file << "Room Number: " << assignedDoctor.roomNumber << "\n";
    pat_file <<
"\n*****\n";
    cout << "\n\n" << assignedDoctor.ward << " Ward Is Allotted And File Are Successfully
Update\n\n";
} else {
    cout << "No Doctor Assign\n";
}

        pat_file.close();
        cout<<"\n\n";
        system("pause");
    system("cls");
        goto b;
    }
}

//---DISPLAY FULL MEDICAL HISTORY OF PATIENT---
if(i==3)
{
    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)
```

[illegible]

[illegible]



}

}

VIII. CONCLUSION

Understanding and using advanced programming ideas in system development using the C++ programming language is critical in a variety of subjects, including the development of a hospital management system.

- Efficiency and performance: A hospital management system must cope with a large amount of data as well as sophisticated procedures.
- Data Structures and Algorithms: C++'s advanced programming ideas allow for the creation of efficient data structures and algorithms. Hospitals create and manage a large amount of data, including patient information, medical history, appointments, and inventories.
- Modularity and Scalability: To suit changing requirements and growing medical practices, hospital management systems frequently require upgrades and expansions.
- Security and Privacy: Hospital management systems handle sensitive patient data such as medical records, personal information, and financial information.
- Resource Management: To achieve smooth operations, hospitals rely on efficient resource management.
- Error Handling and Robustness: Advanced programming principles include tools for handling errors and managing exceptions. To avoid system crashes or inaccurate data processing, it is critical in a hospital management system to handle exceptions and mistakes smoothly.

In conclusion, advanced programming principles are required for creating a hospital administration system in C++. They contribute to the system's efficiency, scalability, security, integration, and robustness. Using these concepts, developers may create a high-performance system that handles complicated activities, effectively manages data, provides privacy and security, communicates with external systems, and adapts to changing healthcare requirements.