

OOP Project Report

Submitted to: Sir Ali Fatmi

Done By:

- ❖ Arooba Moin(20K-0213)
- ❖ Sabah Mawani(20K-0393)

Contents

1. Introduction.....	3
2. Background.....	4
3. Problem Analysis.....	5
4. Methodology.....	7
5. Class Diagram.....	8
6. Implementation.....	9
7. Topics Covered.....	10
8. Results.....	11
9. Conclusion.....	12

Introduction

The program we have built this semester is a 'Hospital Management System'. It's a computerised version of a Front Office Management of a Hospital, we developed a management system which would appear user friendly, & simple for its users.

It deals with patients records, which with ease are stored in their separate files according to their ID names. Before computers, this was all done manually. The purpose of this program is to make lives easier for doctors & the staff throughout the Hospital.

Background

After many days of researching we came down to two categories for our project, namely management systems and games; however, since we had already worked on 2 games in our previous semester we decided to develop a management system this semester.

Now there were a variety of management systems, however we narrowed down to Hospital Management System as we were interested in how Hospitals were managed as traditionally everything was manually done we thought how could we implement this and make it into a program, mainly how the Front Desk manages the patients at the Hospital, and also how the Doctor deals with the patients.

Problem Analysis

The user is given the choice to Login or Exit when the program starts. When they select '1' to Login they are then asked to enter their respective passwords. Now we have created 2 separate Menus. The first Menu is the Nurse's menu and the second Menu is the Doctor's menu. Depending on which type of menu they would like to open, the user will enter the respective password.

Now the User is given 3 attempts to enter their password, when the User enters 'Nurse' as the password correctly, the password is verified and Nurse Menu opens.

The menu gives 4 options to the user.

1. Add New Patient Record
2. View Patient Information
3. Discharge Patient
4. Logout

When the user selects '1' as their choice, the program asks the user to enter the following information, which are then stored in a separate text file.

Patient Name:

Patient Age:

Patient Sex:

Patient Address:

Patient Contact Number:

Patient Blood Group:

When the user selects '2' they can view all the details they have entered in their record. Now when the user selects '3' the user is then asked to enter the ID of the patient they want to discharge. This detail is also stored in the patients file.

Option '4' Logouts the user from the Nurse's Menu.

Now when the User enters the 'Doctor' password it opens the Doctors Menu as followed.

1. Add Patient Diagnosis
2. View Patient History
3. Change Department
4. Log out

When the user selects '1' the user is allowed to add any patient diagnosis, the doctor is also allowed to prescribe medication for the respective patient. This data is stored in the respective Patient ID text file.

When the User selects '2' the Doctor is allowed to view the patient history.

When the User selects '3' the Doctor is allowed to Change the respective department of the patient. From the following choices:

Departments:

1. Cardiology
2. Dentistry
3. Emergency
4. E.N.T
5. Neurology
6. Oncology
7. Physiotherapy
8. Recovery
9. Surgery
10. Urology

These changes are also stored in the respective department textfiles and the main PatientID txt file.

When the User selects '4', the User is logged off from the menu. Then we come back to the main menu, where the user is again asked if they would like to Login or Exit from the program.

Methodology

A methodology is a set of standards and applications that guide you in organizing your projects to ensure their optimum performance. There are many different methodologies that can be adopted when working on a project depending on the budget, team size, flexibility, and timeline. Our group chose the Critical chain project management (CCPM) methodology.

This methodology consists of identifying and scheduling all of the critical tasks that compromise the project, as well as their dependencies. The critical path, longest sequence of critical tasks, can be visualized using PERT and Gantt charts.

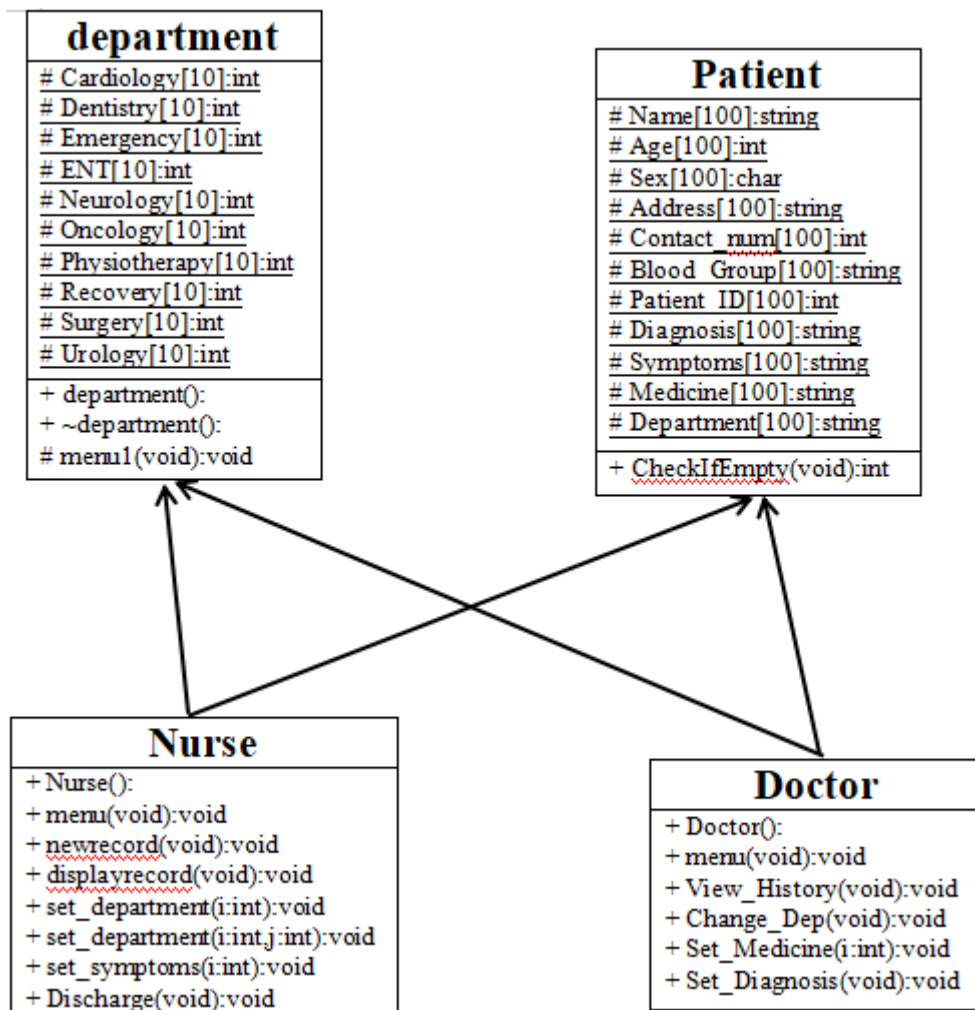
This methodology was perfect for our project since our project had many depending modules, and we needed a visual map of the sequence. We also had a strict deadline and needed buffers to test out our ideas. These conditions were met by the CCPM methodology.

A Gantt chart is a horizontal bar chart developed as a production control tool, it is frequently used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.

Below is the Gantt chart created for the Hospital Management System, done by our group.

Identify Requirements									
Create Class Diagram									
Write the Code									
Add the Data Files									
Testing the Code									
Write Report									
Week	1	2	3	4	5	6	7	8	9

Class Diagram



Implementation

For this project the programming language used is C++. C++ is a powerful general-purpose programming language that supports. It can be used to develop operating systems, browsers, games and so on. C++ supports different ways of programming like procedural, object-oriented, functional, and so on. This makes C++ powerful as well as flexible.

The header files we used in this project are `iostream`, `conio.h`, `fstream`, `string` and `time.h`. The **`iostream`** stands for standard input-output stream. This header file contains definitions to objects like `cin`, `cout`, `cerr` etc. **`Conio.h`** is a C header file used mostly by MS-DOS compilers to provide console input/output. **`fstream`**, It represents both output Stream and input Stream. So it can read from files and write to files. The **`string`** header file introduces string types, character traits and a set of converting functions such as `stoi` which we used. The **`time.h`** header file contains definitions of functions to get and manipulate date and time information.

We used user defined data type Classes & Objects as this was an Object Oriented Programming based Project.

Topics Covered

Encapsulation

Encapsulation is an Object-Oriented Programming concept that binds together the data and functions that manipulate the data, and that keeps both safe from outside interference and misuse. Data encapsulation led to the important OOP concept of data hiding.

Inheritance

Inheritance is used only if a relationship is present between two classes. Inheritance is the technique of deriving a new class from an old one, the capability of a class to derive properties and characteristics from another class. The extended class/child class contains all the features of its base/parent class, and may also have some unique features of its own.

Polymorphism

Polymorphism is a particularly important Object-Oriented Programming concept. Polymorphism occurs when a parent-class reference is used to refer to a child class object, this is one of the most common uses of polymorphism. Furthermore, it's the method in an OOP language that does different things depending on the class of the object which calls it.

File Handling

File Handling is the storing of data in a file using a program. In C programming language, the programs store results, and other data of the program to a file using file handling in C++. Also, we can extract/fetch data from a file to work with it in the program.

Results

When we finally run our code the user is given a choice to either log in or exit. If the user chooses login, the login screen is displayed, and depending on the password entered, the user will either be shown the menu from the Nurse class, or the Doctor class.

If the user enters the password corresponding to the Nurse class, the menu from the Nurse class is displayed. The options displayed are to add a new record, display patient information, discharge a patient, and logout. If the user chooses to add a new record, they are asked to enter the patient's ID, and if the patient is already admitted, the user is told so, else the file is either opened (if it exists) or the file is created. If the user chooses to discharge a patient, the program asks for the patient's ID and then removes the information from the department class and the Patient class. If the user chooses to view a patient's information, all the primary information such as name, age, etc. are shown.

If the user chooses to logout, they are returned to the initial screen where they are given the option to either login, or exit. If the user enters the password corresponding to the Doctor class, then the menu from the class is displayed. The options available are to add patient diagnosis, to view patient history, to change department, or to logout. If the user chooses to add patient diagnosis, then the user is asked to enter the patient's ID, if the patient is admitted then the user can add the diagnosis, and prescribe medicine, else they will be told that the patient is not admitted. If the user chooses to view the patient's history, then the user will be asked to enter the patient's ID, if the patient is admitted then the program will open the corresponding file and display the patients history. If the user chooses to change the department then the user is shown the list of departments and is asked to choose from it.

Conclusion

Our project consists of a secure hospital management system, it has features such as, a proper login, a menu, etc. In comparison to the existing model, we have included a proper menu that will allow the user to choose specific options by checking their security level.

Another feature that has been included is the security of our program, we have set two different passwords for the system, and only if this is entered correctly will the system let the user proceed on. This is a feature which is found to be commonly lacking in most of the systems.

Moreover, our management system includes functions that allow the user to add new records, update previous records and also allow the user to view the patient's previous record history.