

**FUNDAMENTALS IN SOFTWARE DEVELOPMENT**

ASSIGNMENT

PROGRAM DOCUMENTATION

BY:

ABIYYU TAJ MAHASIN BAGINDO  
TP058652

APU1F2002CS(IS)

Table of Contents

[Introduction and Assumptions 3](#_Toc42437596)

[Design 3](#_Toc42437597)

# Introduction and Assumptions

A pandemic has recently struck the world. COVID-19 or commonly known as the Coronavirus is a virus that causes respiratory tract infections. Since the global pandemic started, hospitals have been significantly more active. Patients go in and out everyday, and an information system has to be made to manage all those patients’ data effectively and efficiently.

I have developed a program that allows the user to register patients, test patients, modify those patients’ data, display statistics, and search for the data they want. All the features mentioned and specified in the assignment question are included in the program and are in working order.

The program is event-driven which means the program behaves and react according to the user’s input. The program is menu-driven where the user will use menus and options to navigate through the program. However, the program looks rather dull due to graphics not being a focus during development.

The program assumes the user understands English and inputs the input requested. The program may function not as expected or wildly when the user inputs data that the program was not meant to process. For example, most of the menus require simple single digit inputs such as the number “1” or “2”. Entering the option’s name or “A” instead of “1” will cause the program to not work or function properly.

# Design

The design of the program is quite simple. A main function menu serves as the foundation of the whole program. It simply prints the main menu and awaits user input. It will then call other functions such as registerPatient or getStatPatient depending on the user’s input. Until a user inputs something, the main function menu will not do anything.

The flowchart below shows the flow of the program globally.

A close up of a map

Description automatically generated

Figure 2.1 Global flowchart

This is the flowchart of the function menu(), which allows the user to navigate through the different aspects and utilize the different features of the program. How it works is it displays a menu and prompts the user to enter a number that corresponds to the option they want to choose. That number will then be stored in a variable called option and a conditional part of the program will call a function, exit the program, or return to the beginning of the function depending on the number the user entered.

For example, a user runs the program, the following display will be printed out by the program. The user can then input a number that corresponds to the feature they want to use. For example, if a user wants to register a new patient, the user should simply enter “1” and press the enter key on their keyboard

A screen shot of a social media post

Description automatically generated

Figure 2.2 Main menu

After pressing enter, the program will then prompt the user to enter multiple kinds of patient data such as the patient’s name, the patient’s ID, and the patient’s group. The image below shows a user registering a new patient named Flandre.

A screen shot of a computer

Description automatically generated

The user has successfully registered a new patient using the first module, the register module. After using a module, the program will run the function exitMenu which is basically the program asking the user whether they want to continue using the program or exit the program.

How the register module works is it will print a single line asking the user to enter a piece of data. After the user entered the data, the program will then read the data and assigned it to a variable. Below is a pseudocode for a piece of the register module that is responsible for gathering a patient’s ID

PRINT "Enter patient ID: "

READ patientID

It prompts the user to enter the new patient’s ID, and then stores it inside a variable called patientID. The module will continue to gather data until it has enough and then combines all data gathered and some additional default values to form a patient record. Below is the pseudocode.

patientData = patientName+";"+patientID+";"+patientMail+";"+patientGroup+";"+patientZone+";"+"N"+";"+"N"+";"+"N"+";"+"N"+";"+"0"+";"+"N"+";"+patientCondition+";"+"X"

All data that belongs to the new patient is concatenated with a “;” in between to separate the different data and then assigned as a string to the variable patientData.

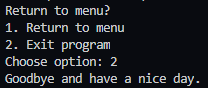


Figure 2.3 the exitMenu function

The program uses a single text file as database to store and fetch patient data before or after processing. The text file is named patient.txt and stores all patient data be it their names, ID, or test results.