



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Summer, Year: 2021), B.Sc. in CSE (Day)

Course Title: Data Structure Lab
Course Code: CSE 106 Section: 213DC

Lab Project Name: Medical Store Mangement System

Student Details

	Name	ID
1.	Naimul Haque Bhuyan Rahat	213902100

Submission Date:

Course Teacher's Name: Md. Sultanul Islam Ovi

[For Teachers use only: Don't Write Anything inside this box]

Lab Project Status

Marks:

Signature:

Comments:

Date:

Table of Contents

Chapter 1 Introduction	3
1.1 Introduction.....	3
1.2 Design Goals/Objective	3
Chapter 2.....	3
Implementation of the Project.....	4
Chapter 3 Conclusion.....	10

Chapter 1

Introduction

1.1 Introduction

Originally, C language is developed from two previous languages, BCPL and B. BCPL which were developed in 1967 by Martin Richards as a language for writing operating systems and compilers. C was evolved from B by Dennis Ritchie at Bell Laboratories and it was implemented in 1972. It initially became widely known as the development language of the UNIX operating system. Lots of today's leading operating systems are written in C and C++. C language is mostly hardware independent as it is possible to write C programs that are portable to most computers.

Why we use C language C has been used successfully for each kind of programming problem thinkable from operating systems to spreadsheets to expert systems - and efficient compilers are accessible for machines ranging in power from the Apple Macintosh to the Cray supercomputers. The largest measure of C's success appears to be based on strictly sensible considerations:

1. The standard library concept;
2. the ease with that applications can be optimized by hand-coding isolated procedures;
3. a powerful and varied repertoire of operators;
4. the portability of the compiler;

1.2 Design Goals/Objective

The objective of this project is to develop a medical store management system for customer and the owner to satisfy them with all effectiveness facilities.

- Ob1: To minimize the number of staff at the medical store.
- Ob2: To enable customer to check the availability of required medicine.
- Ob3: To increase work flow, accuracy and reduce the work amount and time.

A small medical store has just purchased a computer for its new automated management system. The owner has asked to program the new system in C. It is required to write a program to assign the medicine, customer, supplier, price list in the database. If customer search for a medicine and if the system found the medicine then the customer can buy the medicine otherwise the system show a message stock finished.

After the quantity of medicine is finished and the customer want to buy it then the system displaying the medicine stocked is finished and then the customer easily change her/his decision and can go to other shop to buy this medicine without losing time.

Chapter 2

1. Implementation

```

/*
Naimul Haque Bhuyan Rahat
ID: 213902100
*/
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
#include<time.h>

//Structure
typedef struct mdeicine{
    char medicine_name[30];
    char customer_name[30];
    int Medicine_ID;
    int quantity;
    float sale;
    int squantity;
    float total;
    float unit;
    float cost;
    float profit;
}medicine;

// Function Prototype
void purchaseMedicine();
// void saleMedicine();
void searchMedicine();
void stockMedicine();

int main(){
    int num;
    do{
        system("cls");
        printf("\n\n\n");
        printf("t===== Welcome to Medical Store Management System
=====\\n\\n");
        printf("\\t\\t\\t\\t\\t[1]> PURCHASE NEW MEDICINE\\n\\n");
        printf("\\t\\t\\t\\t\\t[2]> SALE MEDICINE\\n\\n");
        printf("\\t\\t\\t\\t\\t[3]> SEARCH MEDICINE\\n\\n");
        printf("\\t\\t\\t\\t\\t[4]> STOCK OF MEDICINE\\n\\n");
        printf("\\t\\t\\t\\t\\t[6]> Exit\\n\\n");
        printf("t=====
=====\\n\\n");
        printf("\\t\\t\\t\\t\\t Enter Your Choice: ");
        scanf("%d",&num);

        switch(num){
            case 1:
                purchaseMedicine();
                break;
            case 2:
                saleMedicine();
                break;
            case 3:
                searchMedicine();
                break;
            case 4:
                stockMedicine();

```

```

        break;
    }
    getch();
}while(num!=6);

    return 0;
}

//Function Definition
void purchaseMedicine(){
    system("cls");
    medicine *s;
    int n, i;

    printf("Enter Medicine Type Amount: ");
    scanf("%d",&n);

    FILE *fp;

    s = (medicine*)calloc(n,sizeof(medicine));

    if(!fp){
        fp = fopen("MedicineDB.txt","w");
    }
    else{
        fp = fopen("MedicineDB.txt","a");
    }
    for(i=0; i<n; i++){

        printf("Enter Medicine ID: ");
        scanf("%d",&s[i].Medicine_ID);
        fflush(stdin);
        printf("Enter Medicine Name: ");
        gets(s[i].medicine_name);
        printf("Unit Cost Tk.: ");
        scanf("%f",&s[i].unit);
        printf("Sale Cost Tk.: ");
        scanf("%f",&s[i].cost);

        printf("Quantity: ");
        scanf("%d",&s[i].quantity);
        printf("\n");
        printf("*****\n");
        printf("Total Cost: %.2f \t\t Total Sale: %.2f\n",s[i].unit * s[i].quantity,s[i].cost*s[i].quantity);
        fwrite(&s[i],sizeof(medicine),1,fp);
    }
    fclose(fp);
}

void saleMedicine(){
    system("cls");
    medicine s1;
    int mno, found=0, num;
    FILE *fp, *fp1;

    fp = fopen("MedicineDB.txt","r");
    fp1 = fopen("temp.txt","w");

    printf("Enter Medicine Id: ");
    scanf("%d",&mno);
    printf("Enter Quantity: ");
    scanf("%d",&s1.squantity);
    printf("\nID\t\t\t\t\tNAME\t\t\t\t\tQUANTITY\t\t\t\t\tPRICE\t\t\t\t\t");
    printf("\n_____");
    while(fread(&s1,sizeof(medicine),1,fp)){
        if(mno == s1.Medicine_ID && s1.quantity > s1.squantity){
            found = 1;
            printf("\n%-15d%-30s%-7.2d%-11.2f\n",s1.Medicine_ID,s1.medicine_name,s1.squantity,s1.cost);
            s1.quantity = s1.quantity - s1.squantity;
        }
    }
}

```

```

    }
    fwrite(&s1,sizeof(medicine),1,fp1);
}
fclose(fp);
fclose(fp1);
if(found){
    fp1 = fopen("temp.txt","r");
    fp = fopen("MedicineDB.txt","w");

    while(fread(&s1,sizeof(medicine),1,fp1)){
        fwrite(&s1,sizeof(medicine),1,fp);
    }

    fclose(fp);
    fclose(fp1);
}
else{
    printf("\nDetails Not Found");
}
}

void searchMedicine(){
    system("cls");
    medicine s1;
    int mno, found=0;
    FILE *fp;

    fp = fopen("MedicineDB.txt","r");

    printf("Enter Medicine Id: ");
    scanf("%d",&mno);
    printf("\nID      NAME      QUANTITY      PRICE      ");
    printf("\n_____");
    while(fread(&s1,sizeof(medicine),1,fp)){

        if(mno == s1.Medicine_ID){
            found =1;
            printf("\n%-15d%-30s%-7.2f\n",s1.Medicine_ID,s1.medicine_name,s1.quantity,s1.cost);
        }
    }
    if(!found){
        printf("\nDetails Not Found");
    }

    fclose(fp);
}

void stockMedicine(){
    system("cls");
    medicine s1;
    FILE *fp;

    printf("\nID      NAME      QUANTITY      PRICE      ");
    printf("\n_____");

    fp = fopen("MedicineDB.txt","r");
    while(fread(&s1,sizeof(medicine),1,fp)){
        printf("\n%-15d%-30s%-7.2f\n",s1.Medicine_ID,s1.medicine_name,s1.quantity,s1.cost);
    }

    fclose(fp);
}

// Binary Search
/*int binarySearch(int key, int arr[]){

```

```

int left=0, right=6, middle;
while(left<=right){
    middle = (left + right)/2;
    if(a[middle]==key){
        //printf();
        return 0;
    }
    else if(a[middle]< key){
        left = middle+1;
    }
    else{
        right = middle -1;
    }
}
printf("Itm not found\n");
}

// Bubble Sort
int bubbleSort(){
    int i, j, swap, temp, size, counter=0;

    for(i=0; i<size-1; i++){
        swap = 0;
        for(j=0; j<size-1-i; j++){
            counter++;
            if(a[j] > a[j+1]){
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
                swap = 1;
            }
        }
    }
}

}*/

```

Screenshots

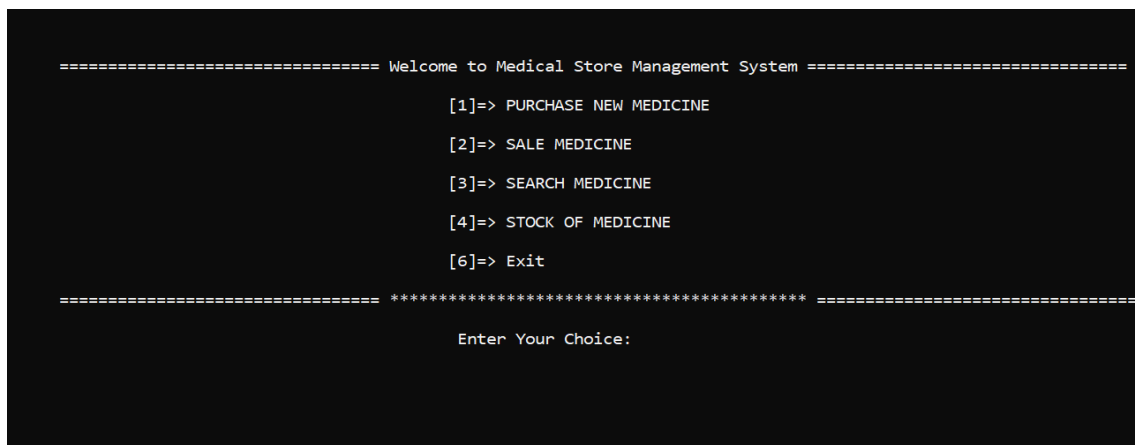


Figure 1 : Main Menu Interface

The user will be directed to the main menu interface. The program is introduced with a few lines of texts. Then four selections are made for the user as the user can choose to reserve, cancel, display or exit the program.

```

Enter Medicine Type Amount: 2
Enter Medicine ID: 1
Enter Medicine Name: Napa
Unit Cost Tk.: 5
Sale Cost Tk.: 7
Quantity: 50

*****
Total Cost: 250.00          Total Sale: 350.00
Enter Medicine ID: 2
Enter Medicine Name: Max Pro
Unit Cost Tk.: 9
Sale Cost Tk.: 13
Quantity: 80

*****
Total Cost: 720.00          Total Sale: 1040.00

```

Figure 2: Purchase New Medicine Function

. If the user choose option one (1), then the user can purchase new medicine for the store.

Enter Medicine Id: 2			
ID	NAME	QUANTITY	PRICE
2	Max Pro	80	13.00

Figure 3: Medicine Search function

If the user choose 3rd option and then use medicine ID then the user can check the availability of the specific medicine.

ID	NAME	QUANTITY	PRICE
1	Napa	50	7.00
2	Max Pro	80	13.00






Figure 4: Stocked Medicine

If the user choose 4th option then user can saw all the stocked medicine in the store.

Thanks For Using Our System

Figure 5: Exit Programm

The user can simply close the program by choose the 6th option from the user menu which is Exit.

 main	9/11/2022 3:00 PM	C Source File	6 KB
 main	9/11/2022 7:43 PM	Application	33 KB
 main.o	9/11/2022 7:43 PM	O File	5 KB
 MedicineDB	9/11/2022 7:35 PM	Text Document	1 KB
 temp	9/11/2022 7:37 PM	Text Document	0 KB

In the program folder the system create 2 text files whis stores the medicine name, id, customer name, medicine price, and the sale information.

Chapter 3

Conclusion

Learning Outcome

The Medical Store Management System is designed for customer to check and buy medicine using the system. A formula is included in the function to calculate the availability of medicine.

From this assignment, I have learnt to implement a few data structure and algorithm concepts in the future projects such as binary search, bubble sort, file handling, arrays, pointers and structures in the program.

Future Scope

- We can use graph theory to include fully automatic staff less system in this project.
- We can add database to our project to store all suppliers and customer information of the Medical store management system.
- We can use the system from anywhere in the world by converting the system online base.

