PROJECT REPORT

Course Code: CSEG1032

Course Title: Programming in C

Project Title: Call Management System

Student Name: Prince Raj Rawat

SAP ID: 590022507

Semester: 1st

Link : https://github.com/Prince-Raj-Rawat/c_project_call_manager.git

1. ABSTRACT

This project implements a menu-driven Call Management System using the C programming language. The program allows the user to add, view, search, and delete contact records stored persistently in a file. It demonstrates the use of structures, file handling, and modular programming techniques to build a scalable and maintainable software system.

2. OBJECTIVE

The objective of this project is to apply fundamental and intermediate concepts of the C language to design a working application. It aims to help store contact information efficiently.

- Use structures to represent complex data.
- Employ functions and modular design for code organization.
- Perform file operations for persistent data storage.

3. PROBLEM DEFINITION

Almost every people require simple tools to maintain contact . Manually handling these records is error-prone and inefficient. This project provides a basic command-line system for managing contact information such as Name, Phone Number, and Email Address in a text file.

4. SYSTEM DESIGN AND ALGORITHM

The system is modular, consisting of multiple source files:

- main.c controls program flow and user interaction.

- call.c handles CRUD (Create, Read, Update/Delete) operations.

Algorithm Flowchart

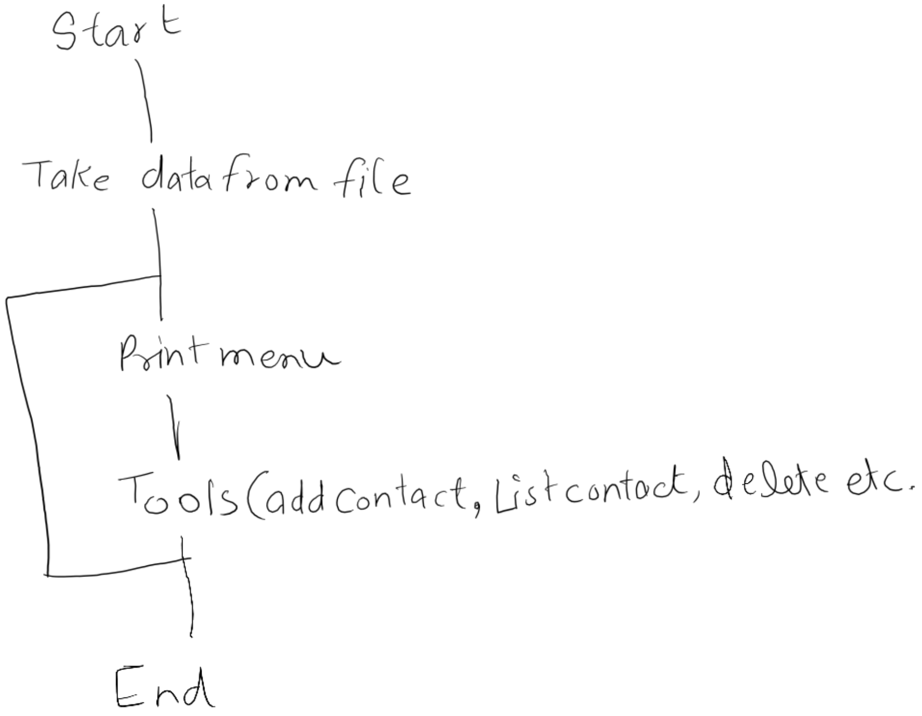

Figure: Program flow for Call Management System

5. IMPLEMENTATION DETAILS

Key Language Features Used:

- Structures to represent contact info.

- File Handling functions (fopen, fwrite, fread, fclose) to store and retrieve data.

- Functions for modularity and clarity.

Example Code Snippet (Writing to File)

```
FILE *fp = fopen("records.txt", "ab");
if (fp) {
    fwrite(&student, sizeof(Student), 1, fp);
    fclose(fp);
}
```

6. OUTPUT (Sample)

Menu:

```
   1. 1.Add Contact
      2. View Contacts
      3. Search Contact
      4. Delete Contact
      5. Exit
   2. INPUT : 1
```

Sample Output

```
Enter Name: John Doe
Enter Phone: 9876543210
Enter Email: john@example.com
Contact added successfully!
```

7. CONCLUSION

The Call Management System demonstrates the practical application of C programming concepts such as structures, file handling, and modular design. It provides a simple yet effective solution for managing contact records in small organizations.

8. FUTURE ENHANCEMENTS

• Add Update Contact feature.

• add grouping in contact like family, friends , classmate etc.

• Create a GUI-based interface for better usability.

9. APPENDIX: main.c (Provided)

```
#include "call.h"

int main(){

    struct contact phone_info[max_contact];
    int count=0;
    int opt=0;

    printf("starting contact manager..
");

    count = upload_from_file(phone_info);
    printf("Loaded %d contacts from file.
", count);

    while(opt != 5)
    {
        menu();

        printf("
enter :");
        scanf("%d",&opt);
        clear_buffer();

        switch(opt){

            case 1 :
                {clear_screen();
                add_contact(phone_info, &count);
                sort_contacts(phone_info, count);
                break;}
            case 2 :
                {clear_screen();
                show_contacts(phone_info, count);
                break;}
             case 3 :
                {clear_screen();
                find_contact(phone_info, count);
                break;}
            case 4 :
                {clear_screen();
                delete_contact(phone_info, &count);
                break;}
            case 5 :
                {save_in_file(phone_info, count);
                break;}
            default:
                {printf("invalid choice please choose no. from 1-5.
");}
        }

    }
    return 0;
```

```
}
```