

C Programming Mini Project Documentation Template

Title Page:

- Project Title : Bloodbank Management System
- Student Name : Dhanushree S
- Register Number : 921724102035
- Department : Computer Science and Engineering
- College Name : Sethu Institute of Technology
- Course : Programming in C
- Guide/Trainer Name : Pugazhmarra
- Submission Date : 25/02/2026

Abstract:

The Blood Bank Management System is developed using C programming language to manage blood donor records efficiently. This system stores donor details such as ID, name, blood group, age, and contact number using structures and file handling. It provides options to add, display, search, and delete donor records through a menu-driven program. The project helps in organizing and maintaining donor information systematically and reduces manual errors.

Introduction:

The Blood Bank Management System is a mini project developed using the C programming language to maintain and manage blood donor records efficiently. The system helps in storing donor details such as donor ID, name, blood group, age, and contact number in a structured format.

This project reduces manual record maintenance and ensures easy access to donor information. It uses file handling to permanently store data and provides a menu-driven interface for user interaction.

Objectives:

- ☐ To maintain donor records in an organized manner
- ☐ To store donor details using structures
- ☐ To implement file handling for permanent data storage
- ☐ To provide options for adding, displaying, searching, and deleting records
- ☐ To reduce manual errors in data management.

Tools and Technology:

- **Programming Language:** C
- **Platform:** Windows
- **Compiler:** Turbo C / Dev C++
- **Concepts Used:**
 - Structures
 - File Handling (fwrite, fread)
 - Functions
 - Menu-driven programming

System Requirements:

Hardware Requirements:

- Minimum 4 GB RAM
- 1 GHz Processor
- Keyboard & Monitor

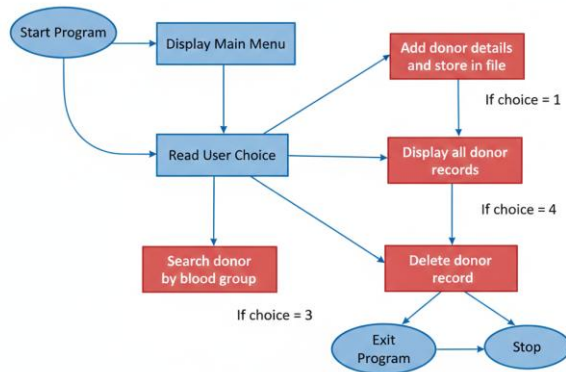
Software Requirements:

- Windows Operating System
- C Compiler (Turbo C / Dev C++)

Methodology / Algorithm:

1. Start the program
2. Display the main menu
3. Read user choice
4. If choice = 1 → Add donor details and store in file
5. If choice = 2 → Display all donor records
6. If choice = 3 → Search donor by blood group
7. If choice = 4 → Delete donor record
8. If choice = 5 → Exit program
9. Stop

Flowchart:



Program Code:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
struct Donor {
```

```
    int id;
```

```
    char name[50];
```

```
    char bloodGroup[5];
```

```

    int age;
    char phone[15];
};

void addDonor() {
    FILE *fp = fopen("bloodbank.dat", "ab");
    struct Donor d;
    printf("\nEnter Donor ID: ");
    scanf("%d", &d.id);
    printf("Enter Name: ");
    scanf(" %[^\n]", d.name);
    printf("Enter Blood Group: ");
    scanf("%s", d.bloodGroup);
    printf("Enter Age: ");
    scanf("%d", &d.age);
    printf("Enter Phone Number: ");
    scanf("%s", d.phone);
    fwrite(&d, sizeof(d), 1, fp);
    fclose(fp);
    printf("\nDonor Added Successfully!\n");
}

void displayDonors() {
    FILE *fp = fopen("bloodbank.dat", "rb");
    struct Donor d;
    printf("\n--- Donor List ---\n");

```

```

while (fread(&d, sizeof(d), 1, fp)) {
    printf("\nID: %d", d.id);
    printf("\nName: %s", d.name);
    printf("\nBlood Group: %s", d.bloodGroup);
    printf("\nAge: %d", d.age);
    printf("\nPhone: %s\n", d.phone);
    printf("-----");
}
fclose(fp);
}

void searchDonor() {
    FILE *fp = fopen("bloodbank.dat", "rb");
    struct Donor d;
    char bg[5];
    int found = 0;
    printf("\nEnter Blood Group to Search: ");
    scanf("%s", bg);
    while (fread(&d, sizeof(d), 1, fp)) {
        if (strcmp(d.bloodGroup, bg) == 0) {
            printf("\nID: %d", d.id);
            printf("\nName: %s", d.name);
            printf("\nAge: %d", d.age);
            printf("\nPhone: %s\n", d.phone);
            printf("-----");
        }
    }
}

```

```

        found = 1;
    }
}
if (!found)
    printf("\nNo donors found with this blood group!\n");
fclose(fp);
}
void deleteDonor() {
    FILE *fp = fopen("bloodbank.dat", "rb");
    FILE *temp = fopen("temp.dat", "wb");
    struct Donor d;
    int id, found = 0;
    printf("\nEnter Donor ID to delete: ");
    scanf("%d", &id);
    while (fread(&d, sizeof(d), 1, fp)) {
        if (d.id != id) {
            fwrite(&d, sizeof(d), 1, temp);
        } else {
            found = 1;
        }
    }
    fclose(fp);
    fclose(temp);
    remove("bloodbank.dat");
}

```

```

        rename("temp.dat", "bloodbank.dat");
    if (found)
        printf("\nDonor Deleted Successfully!\n");
    else
        printf("\nDonor ID not found!\n");
}

int main() {
    int choice;
    while (1) {
        printf("\n===== Blood Bank Management System =====\n");
        printf("1. Add Donor\n");
        printf("2. Display Donors\n");
        printf("3. Search Donor by Blood Group\n");
        printf("4. Delete Donor\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1: addDonor(); break;
            case 2: displayDonors(); break;
            case 3: searchDonor(); break;
            case 4: deleteDonor(); break;
            case 5: exit(0);
            default: printf("Invalid Choice!\n");
        }
    }
}

```



```
    }  
}  
return 0;  
}
```

Sample Input / Output:

Sample Input – 1 (Adding Donor):

===== Blood Bank Management System =====

1. Add Donor
2. Display Donors
3. Search Donor
4. Delete Donor
5. Exit

Enter your choice: 1

Enter Donor ID: 101

Enter Name: Arun Kumar

Enter Blood Group: O+

Enter Age: 25

Enter Phone Number: 9876543210

Donor Added Successfully!

Sample Input – 2 (Display Donors):

Enter your choice: 2

--- Donor List ---

ID: 101

Name: Arun Kumar

Blood Group: O+

Age: 25

Phone: 9876543210

Sample Input – 3 (Search Donor):

Enter your choice: 3

Enter Blood Group to Search: O+

ID: 101

Name: Arun Kumar

Age: 25

Phone: 9876543210

Sample Input – 4 (Delete Donor):

Enter your choice: 4

Enter Donor ID to delete: 101

Donor Deleted Successfully!

Sample Input – 5 (Exit):

Sample Input – 5 (Exit)

Program Terminated.

Result:

The program executed successfully and produced the expected output.

Applications:

- ☐ Used in hospitals to maintain and manage donor records efficiently.
- ☐ Helps in quickly searching required blood groups during emergency situations.
- ☐ Useful in blood donation camps to store and organize donor information.

Conclusion:

The project demonstrates basic C programming concepts.

Future Enhancements:

- ☐ Adding blood stock management to monitor available blood units.
- ☐ Implementing a user login system for better security.
- ☐ Developing a graphical user interface (GUI) for improved usability.

References:

- E. Balagurusamy, “**Programming in ANSI C**”, McGraw Hill Education.
- Yashavant Kanetkar, “**Let Us C**”, BPB Publications.
- Online tutorials and documentation from C programming resources (e.g., GeeksforGeeks).