INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA
CT025-3-2-PCPP
PROGRAMMING CONCEPTS IN C++
UC2F1704EE(IT)\UC2F1704IS\UC2F1704IT\UC2F1704IT(MBT)\UC2F 1704SE

NAME: SHAMEE MRIMMOAEE AHMED

TP NUMBER: TP040804

PROJECT TITLE: X-TREME BLOOD BANK MANAGEMENT SYSTEM

LECTURER NAME: CHANDRA REKA A/P RAMACHANDIRAN

HAND OUT DATE: 30 OCTOBER 2017

HAND IN DATE: 22 January 2018

WEIGHTAGE: 50%

INSTRUCTIONS TO CANDIDATES:

1. **Submit your assignment at the administrative counter**
2. **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)**
3. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld**
4. **Cases of plagiarism will be penalized**
5. **The assignment should be bound in an appropriate style (comb bound or stapled).**
6. **Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**
7. **You must obtain 50% overall to pass this module.**

Table of Contents

1.0 Introduction

1.1 System Description

The name of this system is X-treme Blood Bank Management System, this software is developing for collecting blood through blood donor and supply blood to the hospital. There is three type of user one is an admin, who control most of the function of this software and the other two users are hospital and blood donor. Admin needs to register hospital and blood donor to access the system. Moreover, admin can view all the register user in the system. Beside that admin also register blood donation camp so that blood donor can view the nearest blood donation camp and donate blood. Hospital user uses this system for requesting specific blood group and then an admin needs to accept their request. Beside that hospital user and blood, the donor can view their profile and update if needed.

This system is work automatically that is when admin accepts the request from the hospital, then the bloodstock is automatically decrease based on the number of unit blood hospital request. Moreover, when blood donor request for a donation of blood then the bloodstock is automatically increase based on the unit of specific blood group that donor will be donating.

1.2Assumption

1.Admin can only register user and view the number of user register into the system but admin does not have the permission to delete or update user

2.Admin can only accept the request of the hospital.

3.Hospital and Blood donor can only use this system for requesting blood group and donating blood.

2.0 Design of the solution

2.1 Use Case Diagram

Figure 1: Use Case Diagram

2.2 Use Case Description

Use case name	Register Hospital / Blood Donor
Summary	**Admin register hospital and blood donor.**
Dependency	**No**
Actor	**Admin**
Precondition	**Admin has selected register hospital or blood donor.**
Main sequences	**1. Admin has selected register new hospital user or blood donor** **2. Admin provides the information of the registered user to the system.** **3. System will display a message that new user added to the system.**
Alternative sequences	
Postcondition	**A new user is added to the system.**

Use case name	Register Blood Donation Camp
Summary	**Admin register Blood Donation Camp.**
Dependency	**<<Extend>> Update camp**
Actor	**Admin**
Precondition	**Admin has selected register blood donation camp.**
Main sequences	**1. Admin has selected register new blood donation camp.** **2. Admin provides the information of the blood donation camp to the system.** **3. System will display a message that new blood donation camp added to the system.**
Alternative sequences	
Postcondition	**A new user is added to the system.**

Use case name	Manage Blood Stock
Summary	**Admin can add and view the bloodstock.**
Dependency	**No**
Actor	**Admin**
Precondition	**Admin has selected add bloodstock otherwise admin selected view bloodstock.**
Main sequences	**1. Admin has selected add bloodstock to add new blood group to the system otherwise admin can view the number of blood available to the system.** **2. System will display all the record of bloodstock.**
Alternative sequences	
Postcondition	**A new blood group added to the system.**

Use case name	Manage Hospital Request
Summary	**Admin can view the request from the hospital and accept or reject the request.**
Dependency	**No**
Actor	**Admin**
Precondition	**Admin has selected manage hospital request.**
Main sequences	**1. Admin can view the number of request from the hospital and then type the hospital name, blood group and number of blood unit to accept the request.** **2. System will display a message that request is accepted.**
Alternative sequences	**Admin can view the hospital request for blood.**
Postcondition	**Decrease the total number of blood unit from bloodstock.**

Use case name	Login
Summary	**Admin, hospital user and blood donor provide their username and password to access the system**
Dependency	**No**
Actor	**Admin, Hospital and Blood Donor**
Precondition	**User provides the username and password**
Main sequences	**If username and password are correct then the system will show login successfully otherwise the system will ask to provide username and password again.**
Alternative sequences	
Postcondition	**System will display user menu.**

Use case name	Update Profile
Summary	**Hospital and Blood Donor can update their information.**
Dependency	**No**
Actor	**Hospital and Blood Donor**
Precondition	**User has selected view profile.**
Main sequences	**1. User needs to choose which information they want to update. 2. System will show update successfully.**
Alternative sequences	**Hospital and blood donor can view their profile.**
Postcondition	**New information about the user added to the system.**

Use case name	Request for Blood
Summary	**Hospital user uses this system for requesting a specific blood group and the number of blood unit.**
Dependency	**No**
Actor	**Hospital**
Precondition	**User has selected request for blood.**
Main sequences	**1. User need to provide the name of blood group and the number of blood unit. 2. System will show request submitted.**
Alternative sequences	
Postcondition	**Request send to the admin.**

Use case name	View Processing Blood Request
Summary	**Hospital user can view the number of blood request is still processing.**
Dependency	**No**
Actor	**Hospital**
Precondition	**User has selected view processing blood request.**
Main sequences	**1. User can view the number of blood request is still processing 2. When admin accepts request then the blood request is deleted from the system.**
Alternative sequences	
Postcondition	**Hospital user can again request for the blood.**

Use case name	Request for Blood Donation
Summary	**Donor can use this system to request for blood donation.**
Dependency	**No**
Actor	**Blood Donor**
Precondition	**User has selected request for blood donation.**
Main sequences	**1. User can view the number of blood camp name and their location.** **2. User selects the blood donation camp and enter the blood unit, they want to donate.** **3. System will give a message that request accepted.**
Alternative sequences	**Blood donor can view the blood donation camp record.**
Postcondition	**Increase the total number of blood unit from bloodstock.**

2.3Class Diagram

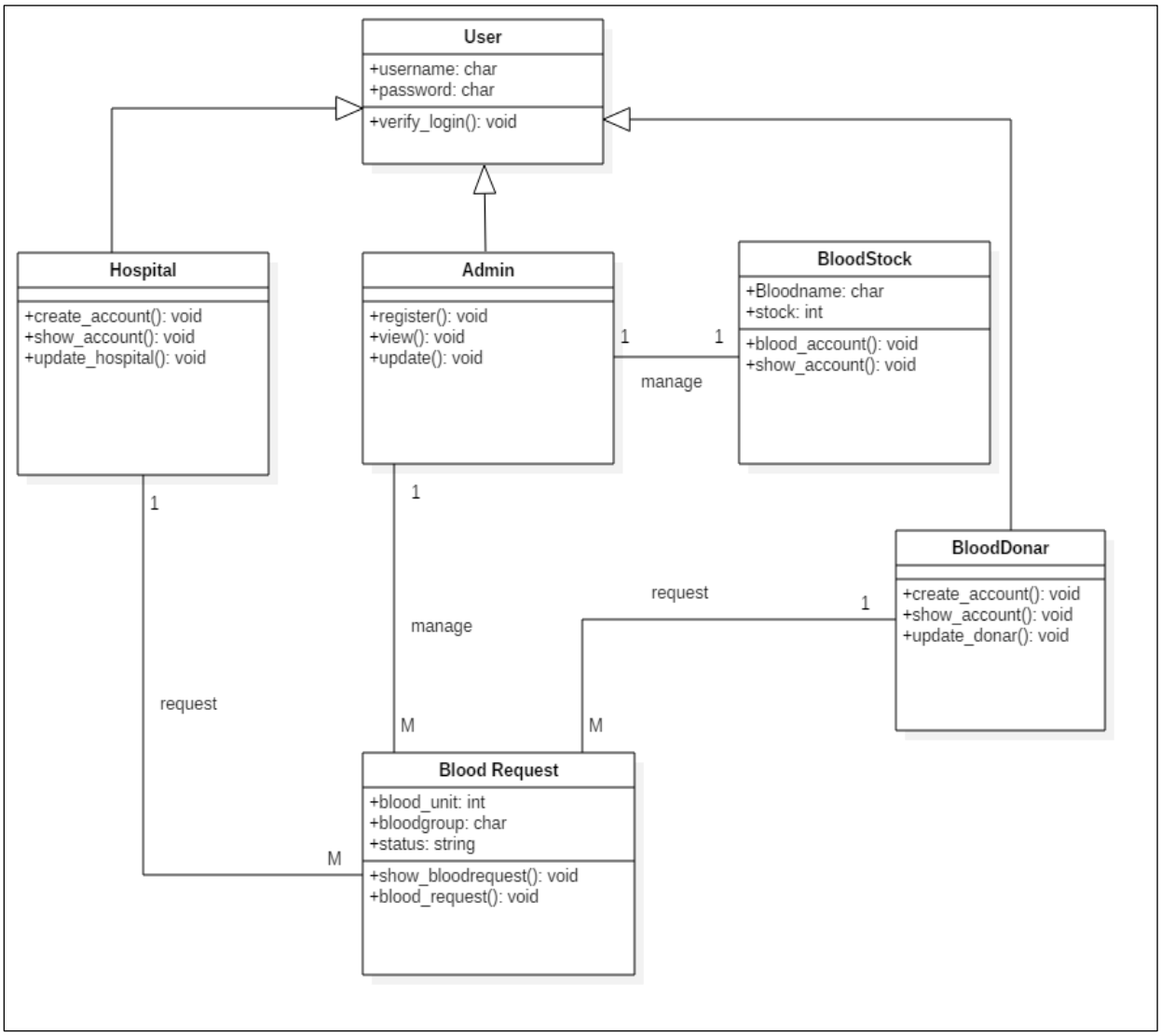

Figure 2: Class Diagram

3.0Explanation of Code

3.1Classes

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include "stdafx.h"
#include <string>


using namespace std;

class BloodStock
{
public:
    char Bloodname[10];
    int stock;

    void blood_account();
    void show_account();

};

void BloodStock::blood_account()
{
    cout << "\n=======================|| ENTER BLOOD GROUP INFORMATION ||=======================" << endl;

    cout << "\n\nEnter the name of blood group : ";
    cin >> Bloodname;
    cout << "\n\nEnter the number of stock : ";
    cin >> stock;
    cout << "\n\n\n New Blood Stock Added :";
}

void BloodStock::show_account()
{

    cout << setw(10) << Bloodname << setw(19) << stock << "\n";

}
```

Figure 3: Class

The class can be defined by a template that describes the behavior of the object that supports its type. The class can contain three type of variable that is a local variable that is defined inside the method, instance variable that is variables within the class but outside the methods and the class variable that are variable also within a class but outside the methods that are declared with the static keyword. Besides this class have a number of methods to access the value of a different kind of method. In the above example blood_account() and show_account() are the method.

3.2Inheritance

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include "stdafx.h"
#include <string>
#include "User.h"



using namespace std;

class Hospital : public User
{
public:

    char bloodgroup[20];
    int unit;
    string status;
```

Figure 4: Inheritance

Inheritance allows one class to inherit the feature of another class that is field or method. The class that feature is inherited to another class are called super class like in this example User is the super class and the class that is Hospital, that inherit the feature of User are called sub class.

3.3 File Processing

```
char n[3];
char k[10];
char c;
int flag = 0;
fstream f1;
fstream f2;
int m;
fstream fp, f;
int found = 0;
ifstream in("bloodcamp.txt");
if (!in)
{
    cout << "File not Found";
}
cout << "\n\t\t\t                          *Blood Donation Camp Information List*  \n\n";
cout << "|---Camp Name---|" << setw(10) << "|---Location---|\n";
cout << "-------------------------------------------------------------\n";
while (in.read((char*)&y, sizeof(Blooddonationcamp)))
{
    y.show_account();

}

in.close();

cout << "\n\n\tDo You Want To Donate Blood (y/n?)";

cin >> c;
if (c == 'y' || c == 'Y')
{
    cout << "\n\n\tEnter the name of Blood Camp: ";
    cin >> k;

    cout << "\n\n\tEnter the name of blood group:";
```

Figure 5: File Processing

This code is used for showing the blood camp to the blood donor and accepting the request of donating blood. Here fstream is used for both read and write from the file.

3.4 Global Variable

```
//=========================================|| Global Variable ||=====================================
//=================================================================================================

char current_user[10]; // global variable of user that call outside the function
char user[10];
char bloodName[10];
```

Figure 6: Global Variable

A variable that declared outside the function is called global variable. A global variable is visible until the end of the file in which they are declared. Here, a global variable is used to identify the login username.

3.5 Pointer

```
void Allhospitalinfo()
{
    ifstream in("hospital.txt");
    if (!in)
    {
        cout << "File not Found";
    }
    cout << "\n\t\t\t                          *Hospital Information List*  \n\n";
    cout << "|---Hospital Name---|" << setw(10) << "|---Password---|\n";
    cout << "-----------------------------------------------------\n";
    while (in.read((char*)&h, sizeof(Hospital)))
    {
        h.show_account();

    }

    in.close();
}
```

Figure 7: Pointer

This code uses for viewing the whole hospital record. Here pointer is used that takes the value and convert to float but pretending there is char data rather than float data.

3.6 Compare String

```
while (in.read((char *)&b, sizeof(BloodDonar)))
{
    if (strcmp(b.name, n) == 0 && strcmp(b.password, p) == 0)
    {
        strcpy_s(user, b.name);
        flag = 1;
        cout << "\n\tWELCOME  " << b.name << "\n";
        Donor_Menu();

    }
}
```

Figure 8: String Comparison

Here strcmp compare the two characters until a terminating character is reached. It uses the login function so that it can compare the two-char value and identify that they are a match or not.

3.7 Read char array

```
void hospital_login()
{
    char n[25], p[5];
    int flag = 0;
    ifstream in("hospital.txt", ios::in | ios::out);

    cin.get();
    cout << "\n\n Enter the Hospital Name: ";
    cin.getline(n, 25);
    cout << "\n\n Enter the Password: ";
    cin >> p;
```

Figure 9: Read char

Here cin.getline is used to read the char array and proceed to another line. So, that it identifies the username and password matches or not.

4.0Explanation of Snapshot of the solution output

4.1 Login Menu of User

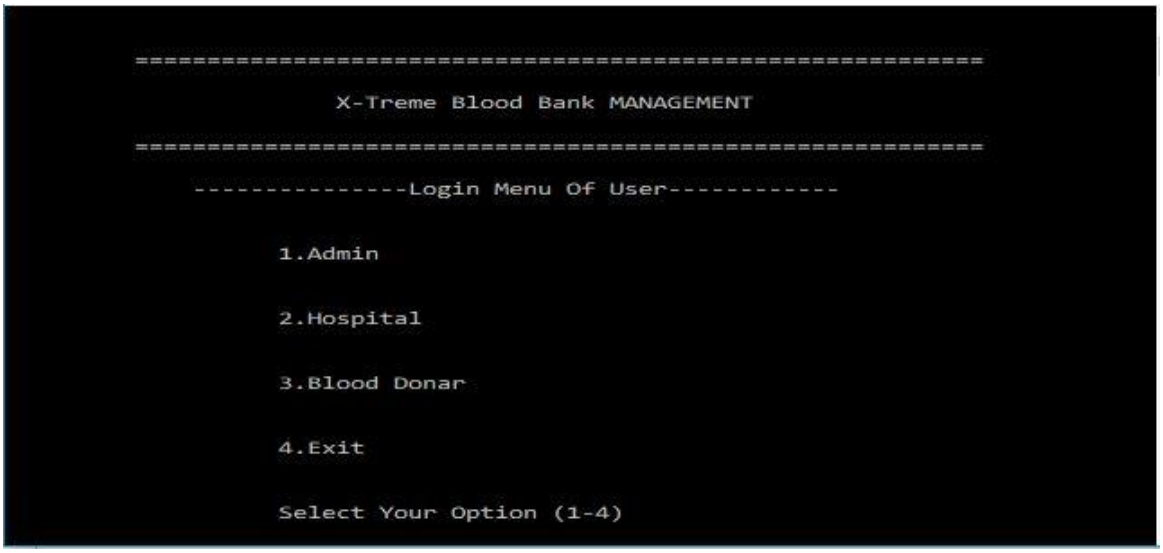

Figure 10: Login Menu

When the software open, it will show the login menu of the user to choose their type otherwise they can exit from the system.

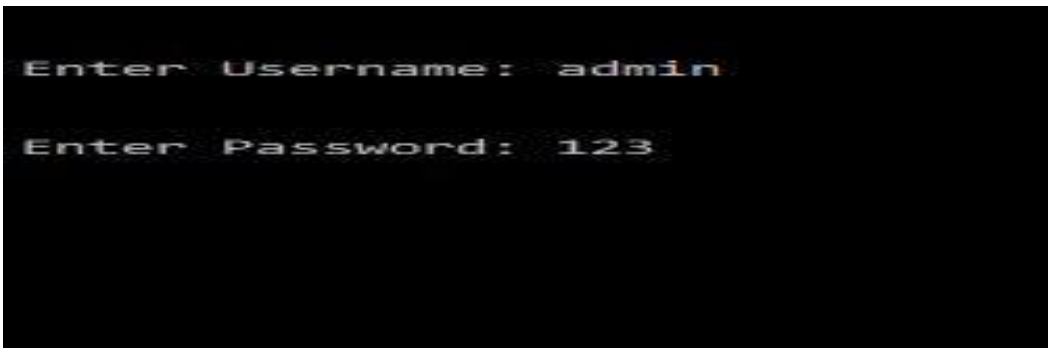

Figure 11: Login Information

User needs to type their username and password to login into the system.

4.2 Admin Menu

Figure 12: Admin Menu

Admin can register blood donor, hospital, blood donation camp and add bloodstock. Admin can also view the hospital, blood donor, bloodstock and blood donation camp record. Moreover, admin uses this system to manage hospital request for blood. Since admin has one update function that is blood camp record.

4.3 Register Blood Donor

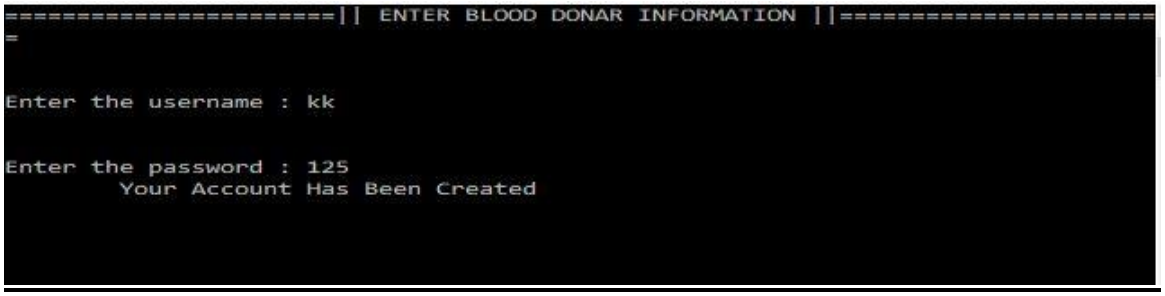

Figure 13: Blood Donor

Admin needs to provide the blood donor name and password to create the account.

4.4 Register Hospital

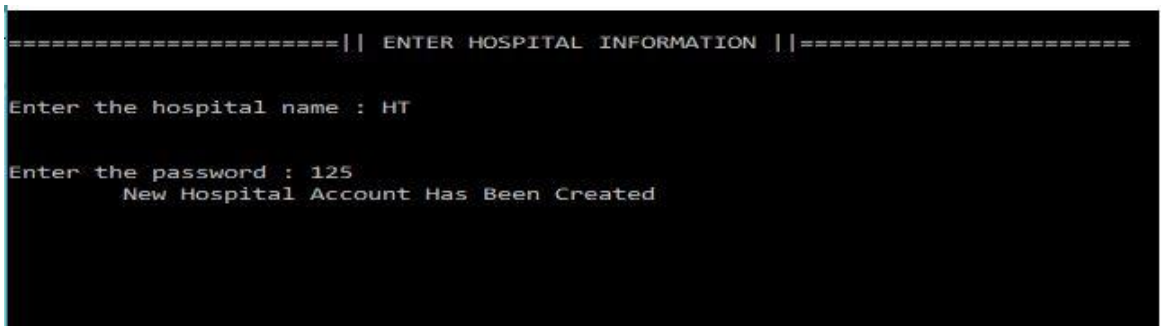

Figure 14: Hospital

Admin needs to provide the hospital name and password to create the account.

4.5 Register Blood Donation Camp

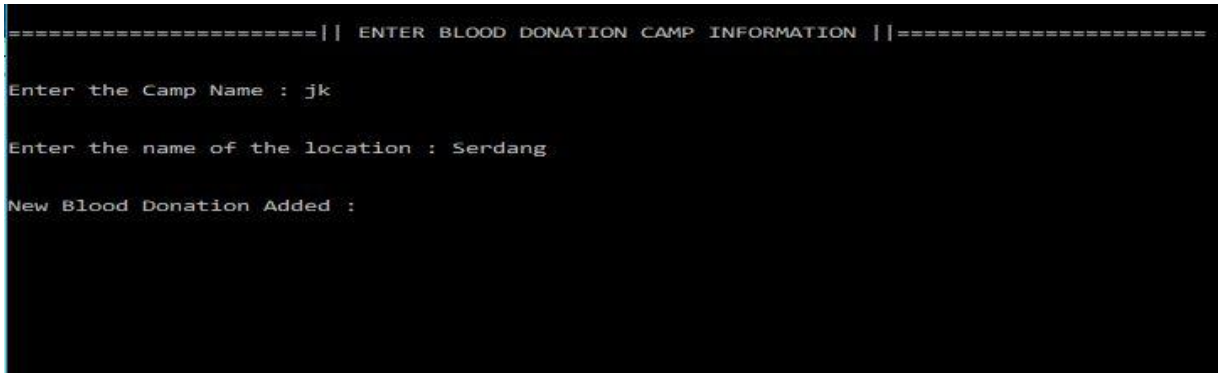

Figure 15: Blood Camp

Admin needs to provide the blood donation camp name and location to create the account.

4.6 Add New Blood Stock

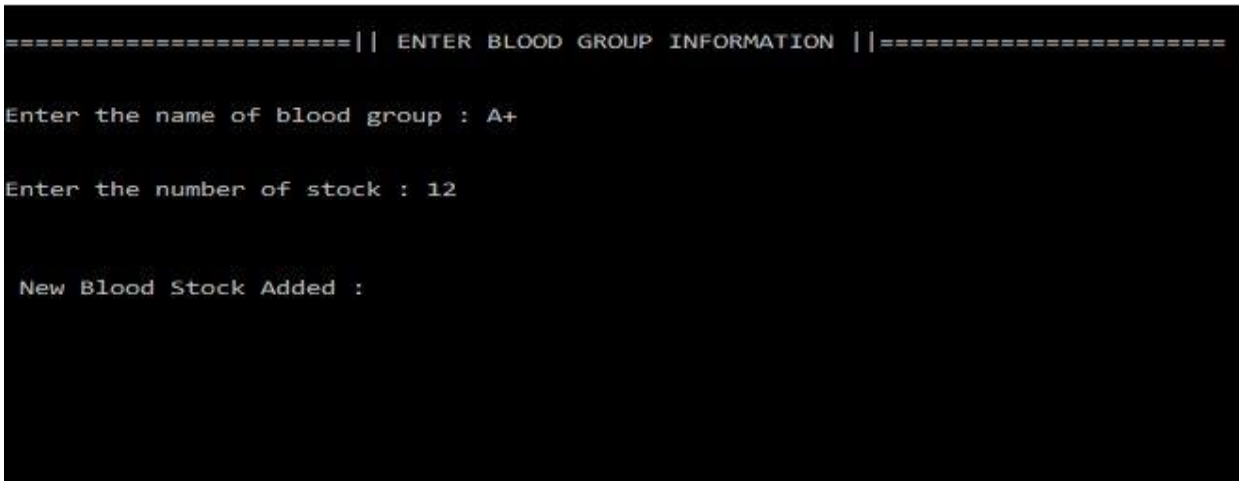

Figure 16: Blood Stock

Admin needs to provide the blood group name and number of stock for creating the account.

4.7 Manage Hospital Request

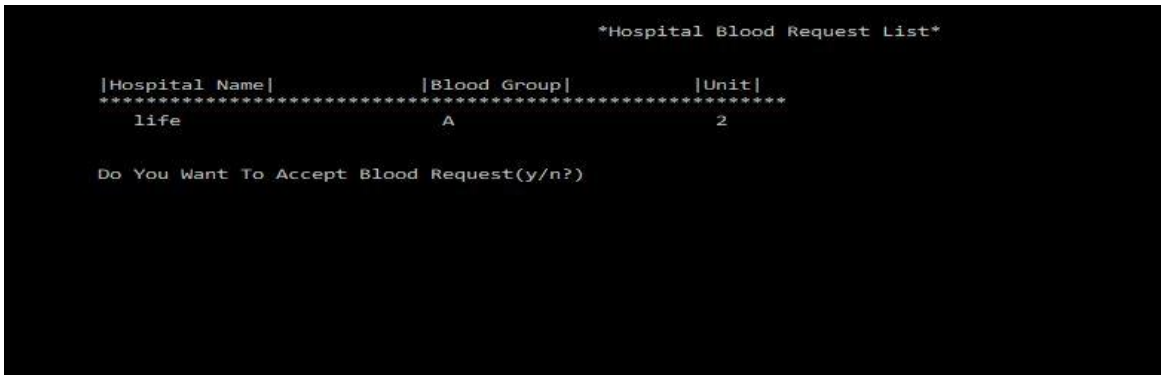

Figure 17: Hospital Request

Admin can view the hospital request and accept or reject the request.

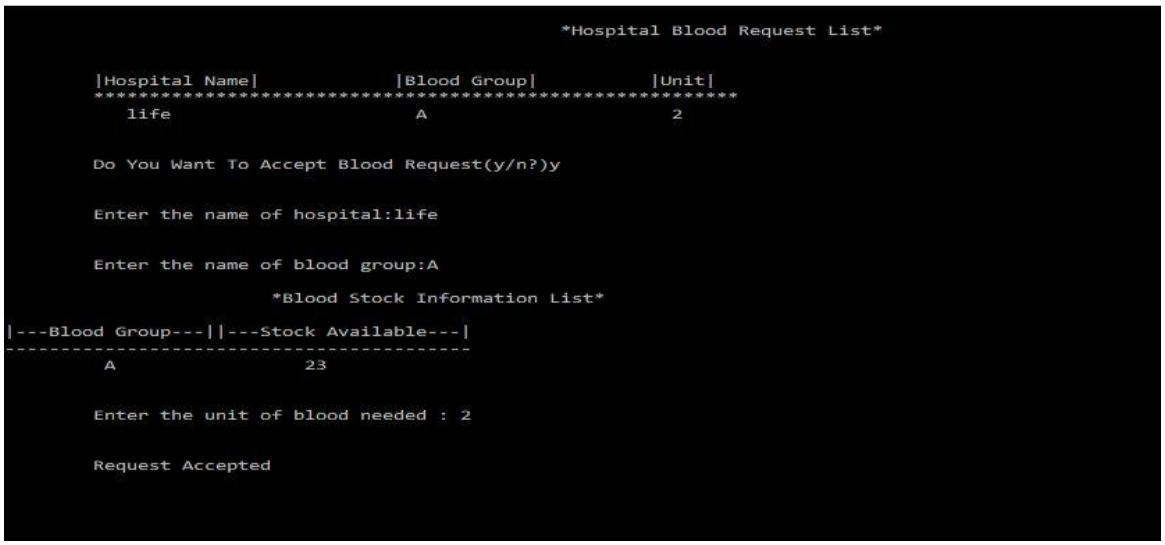

Figure 18: Accept Hospital Request

When admin accepts the hospital request, the system will ask hospital name, blood group and show the available bloodstock. Then admin needs to provide the number of blood unit needed and then the request accepted.

4.8 Blood Donor Information List

Figure 19: View Blood Donor

Admin can view the registered list of the blood donor.

4.9 All Hospital List

Figure 20: View Hospital List

Admin can view the registered list of hospital user.

4.10 All Blood Donation Camp

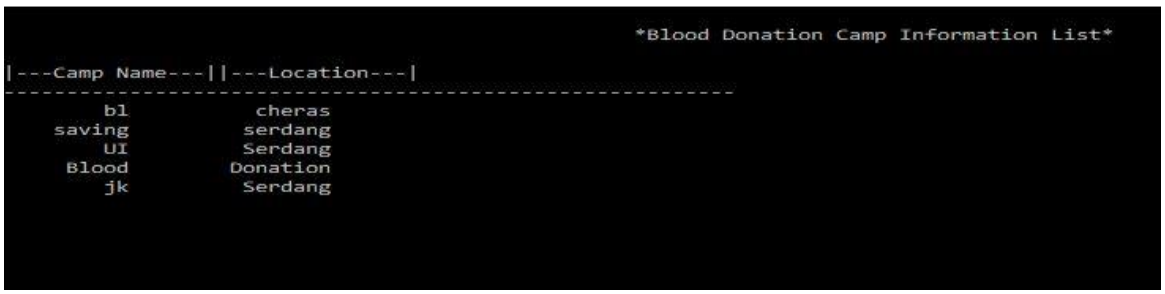

Figure 21: View Blood Donation Camp List

Admin can view the registered list of blood donation camp.

4.11 All Blood Stock Record

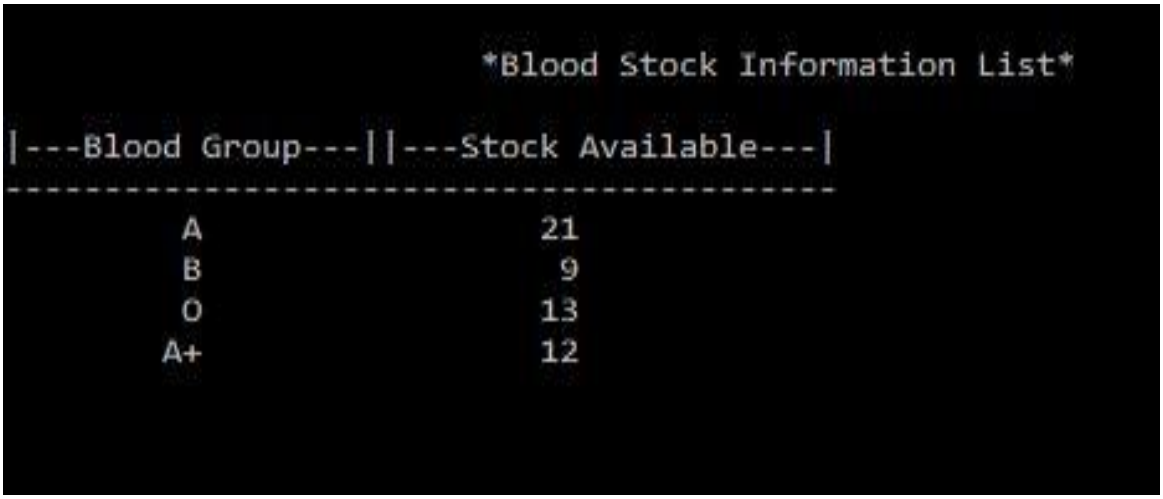

Figure 22: View Blood Donation Camp List

Admin can view the list of bloodstock.

4.12 Update Blood Donation Camp Record

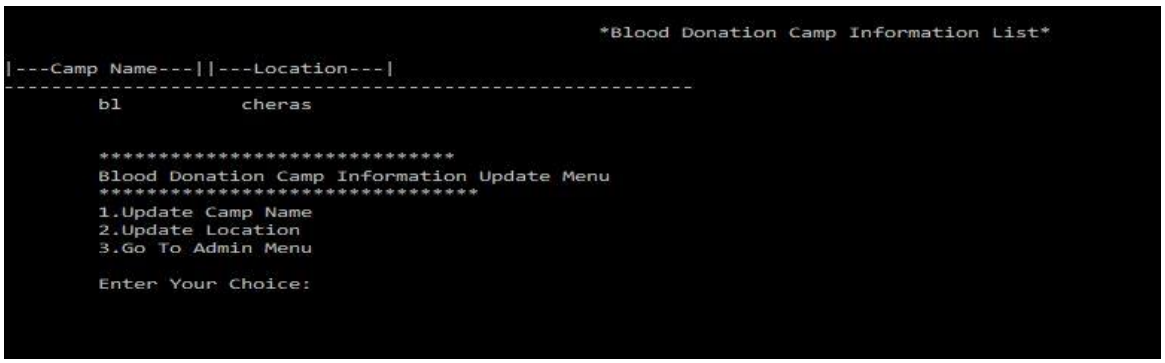

Figure 23: View Blood Donation Camp List

Admin can choose from 1 to 2 to update the blood donation camp record.

4.13 Hospital Menu

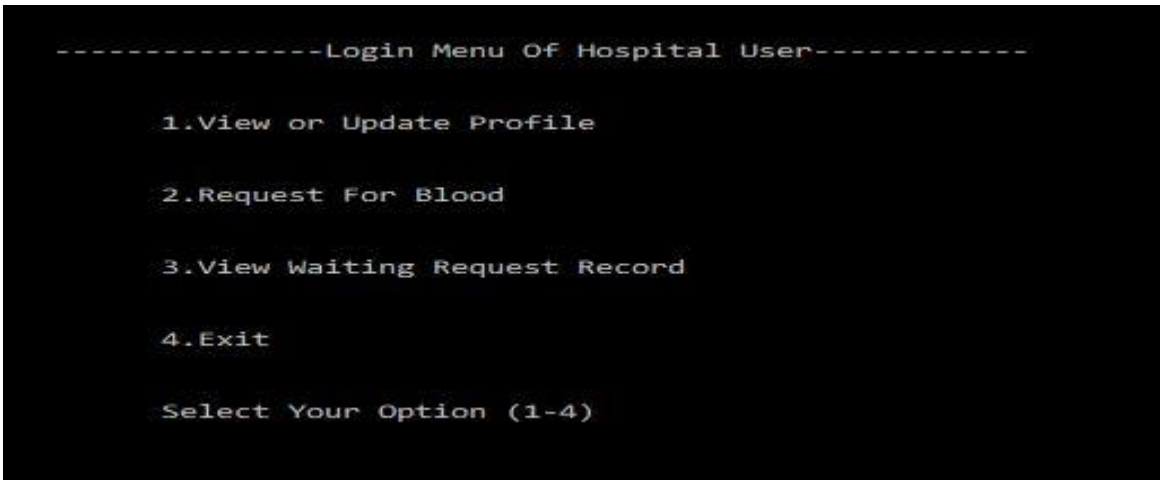

Figure 24: Hospital Menu

This is the hospital menu. Hospital user can view their profile and update if needed. They can request for the blood and see the request that is still processing.

4.14 Hospital Update Menu

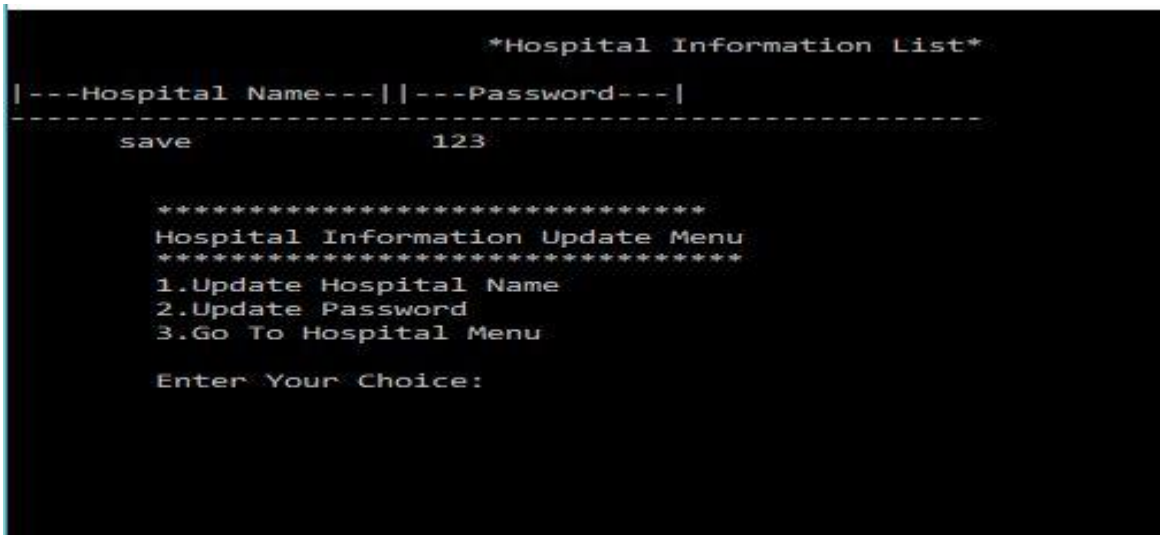

Figure 25: Hospital Menu

Hospital user can view their profile and choose from 1 to 2 to update their profile

4.15 Request for Blood

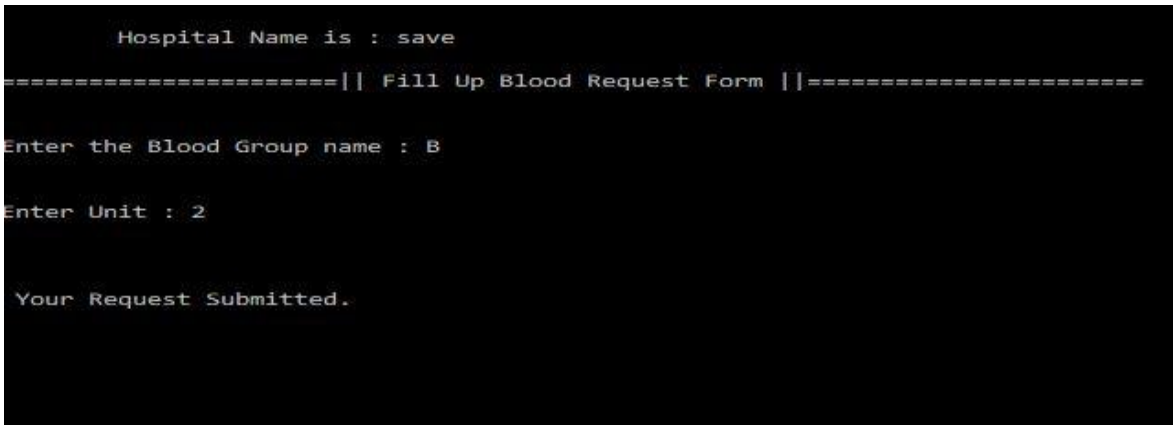

Figure 26: Blood Request

Hospital user needs to provide the name of blood group and the number of blood unit for submitting the request. Hence, each hospital name will automatically add to the request.

4.16 Waiting Blood Request Record

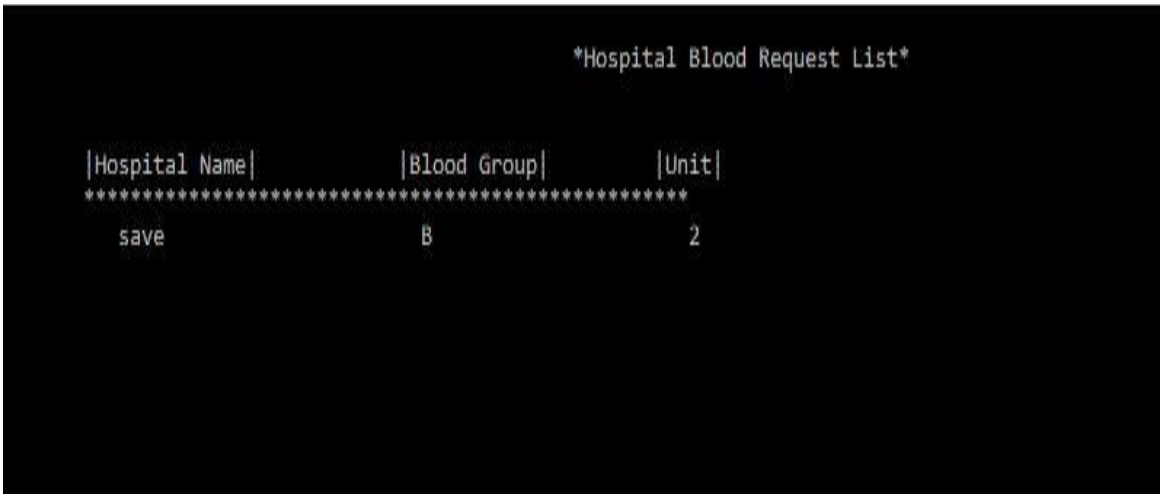

Figure 27: Waiting Blood Request Record

Here, hospital user can view their processing blood request. When the blood request is accepted by an admin, it will automatically delete from this page.

4.17 Blood Donor Menu

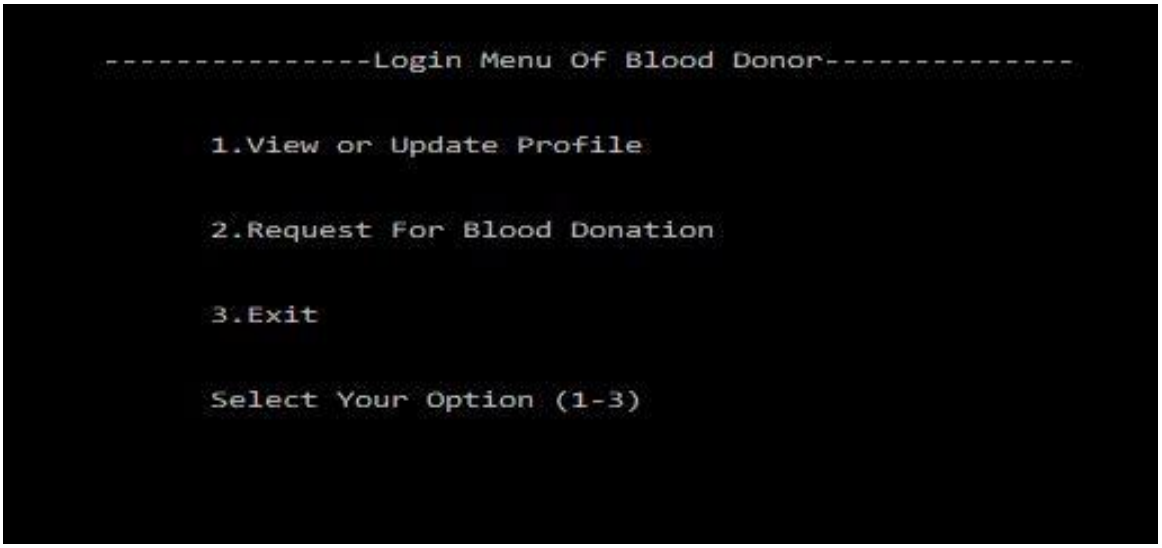

Figure 28: Blood Donor Menu

Blood donor can view and update their profile. Besides that, they can also request for blood donation.

4.18 Blood Donor Update Menu

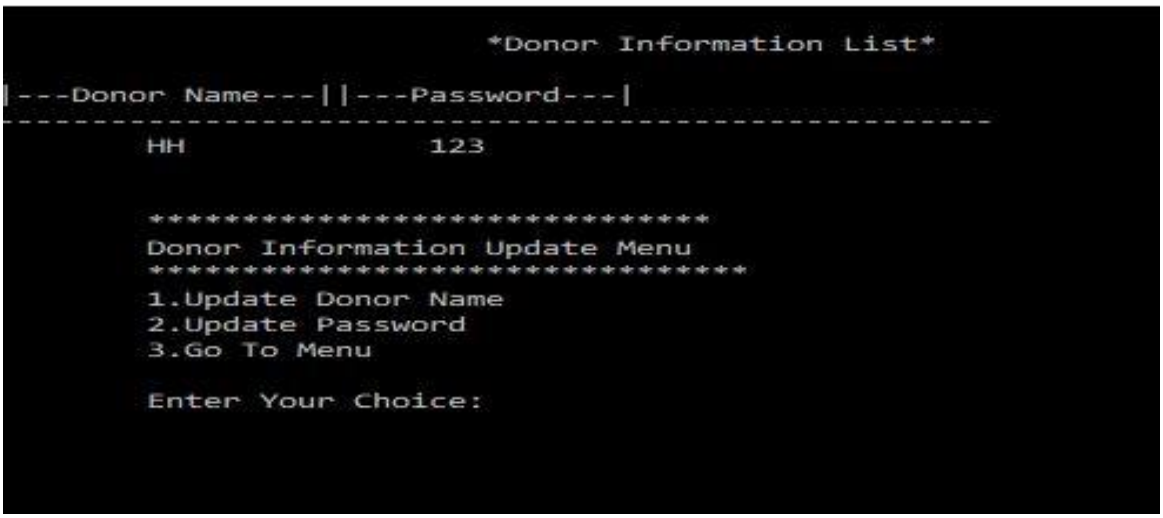

Figure 29: Blood Donor Profile

Blood donor can view their profile and choose from 1 to 2 to update their profile

4.19 Request for Blood Donation

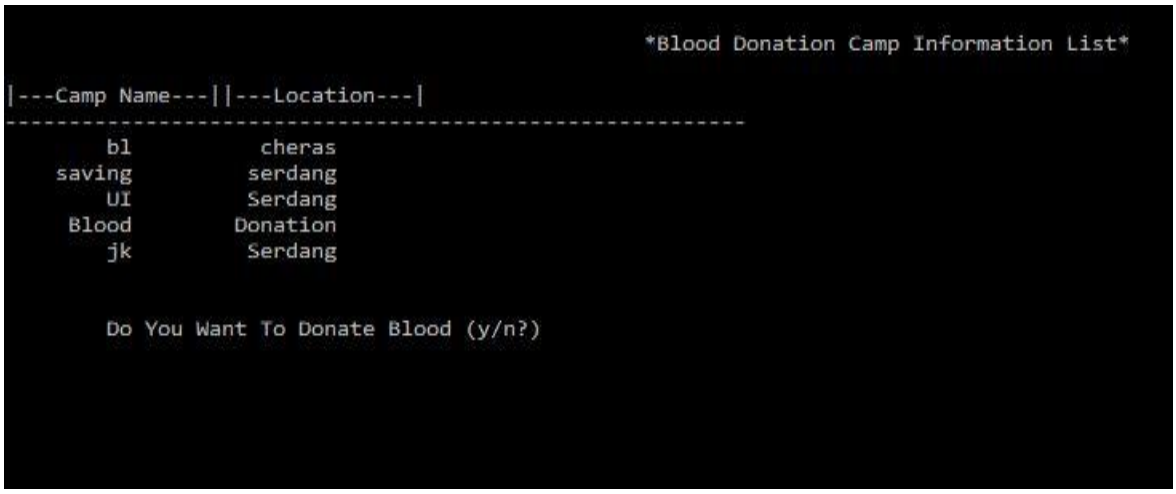

Figure 30: Blood Donation Request

Blood Donor can view the list of blood donation camp name and their location.

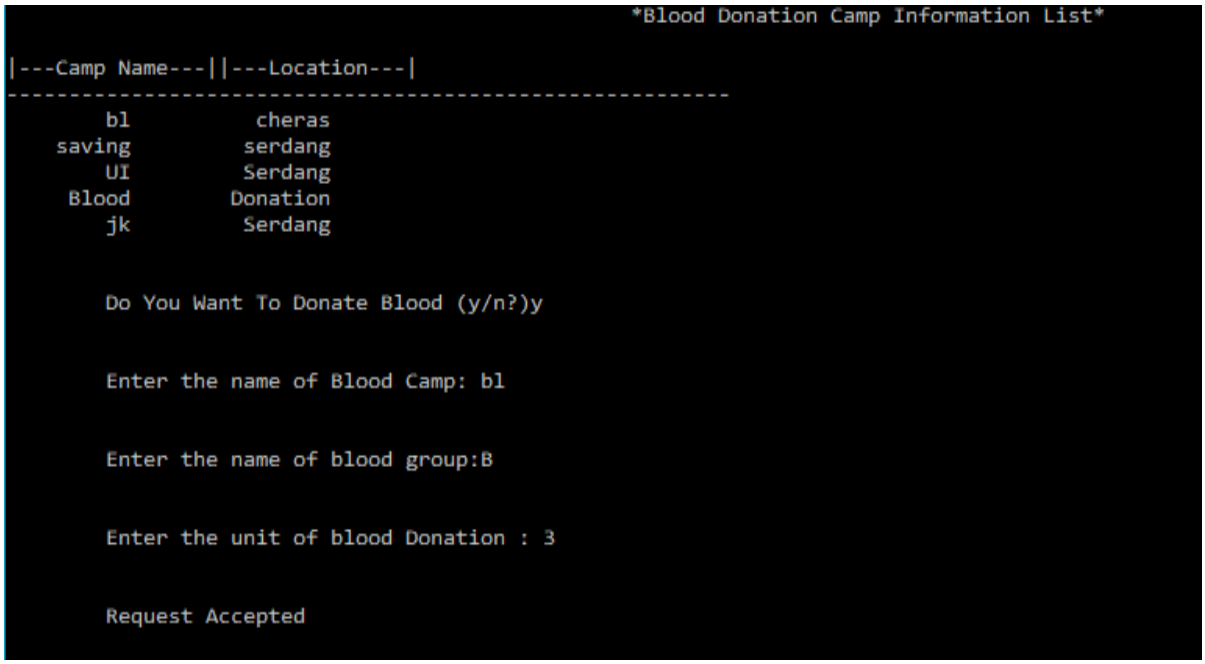

Figure 31: Blood Donation Request

Blood donor needs to provide the name of blood camp, blood group, and a number of blood unit they want to donate then blood donation request is accepted by the system.

5.0 Conclusion

As a concluded from the above discussion that a secure, reliable, faster and an efficient system has been developed that replacing the manual work. This system is highly flexible and reliable to use as well as it will give the accurate detail of the record that save into the system. This system can automatically increase and decrease the bloodstock based on the hospital request and blood donation acceptance.

Since it will save time, the amount of work and reduces paper-based work that will reduce the environmental impact. Moreover, this system is provided to meet expected result for that it is developed but still there is some function that needs to be improved

6.0 References

1.Physics.utah.edu. (2018). *ifstream and ofstream methods*. [online] Available at: http://www.physics.utah.edu/~detar/lessons/c++/canned_classes/node9.html [Accessed 22 Jan. 2018].

2.Learn C++. (2018). *4.2 — Global variables and linkage*. [online] Available at: http://www.learncpp.com/cpp-tutorial/42-global-variables/ [Accessed 19 Jan. 2018].

3.Cplusplus.com. (2018). *strcmp - C++ Reference*. [online] Available at: http://www.cplusplus.com/reference/cstring/strcmp/ [Accessed 20 Jan. 2018].

4.www.tutorialspoint.com. (2018). *Polymorphism in C++*. [online] Available at: https://www.tutorialspoint.com/cplusplus/cpp_polymorphism.htm [Accessed 22 Jan. 2018].

5.Java2s.com. (2018). *Using getline() to read char array : cin getline « Console « C++*. [online] Available at: http://www.java2s.com/Code/Cpp/Console/Usinggetlinetoreadchararray.htm [Accessed 18 Jan. 2018].

6.Cplusplus.com. (2018). *C++ Language - C++ Tutorials*. [online] Available at: http://www.cplusplus.com/doc/tutorial/ [Accessed 18 Jan. 2018].

7.Cppforschool.com. (2018). *C++ Inheritance Question*. [online] Available at: http://www.cppforschool.com/assignment/inheritance_1.html [Accessed 19 Jan. 2018].