Online Blood Donation System

Topic : Online Blood Donation System (OneBlood)

Group no : MLB_08.01_03

Campus : Malabe

Submission Date : 2022.05.17

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21161056	Peiris M.M.A.E.	0710624368
IT21160820	Rupasinghe Y.S.	0717734747
IT21162732	Jayathunge K.A.D.T.R.	0763121956
IT21160370	Dissanayake M.K.S.E.	0718923410
IT21162664	Weerasinghe W.P.D.J.N.	0713007363

Requirements Analysis

Nouns
Verbs

As a Donor

I want to register to the system so that I can get Donor id, so that I can submit my personal details such as name, address, NIC, Gender, DOB, email, phone number, Blood group and password.

I want to login to the system with my username and password, so that I can access the system

I want to update my personal details

I want to make and cancel appointments (appointmentsNO, location, appointment date, so that I can visit the blood donating campaigns and donate blood

As an Assistant (name, address, phone number, email)

I want to login to the system with using my admin username and password

I want to organize campaign with event no, location and event date

I want to make budget report so that I can submit it to the manager

As a Nurse (name, address, phone number, email)

I want to login to the system with using my admin username and password

So that I can check appointments and send confirmation to the assistant

I want to update health report

As a hospital

I want to place blood orders with OrderNO, OrderDate, so that I can refill blood requirement

Noun Verb Analysis

Donor	class
system	outside scope of system
Donor id	attribute
personal details	meta language
name	attribute
address	attribute
NIC	attribute
Gender	attribute
DOB	attribute
email	attribute
phone number	attribute
Blood group	attribute
password	attribute
system	outside scope of system
username	attribute
password	Redundant
system	outside scope of system
personal details	meta language
appointments	Class
appointmentsNO	attribute
location	attribute
appointment date	attribute
blood donating campaigns	class
blood	outside scope of system
Assistant	class
system	outside scope of system
username	Redundant
password	Redundant
appointments	Redundant
donors	Redundant
Campaign	Redundant
event no	attribute
location	Redundant
event date	attribute
budget report	Class(boundary)
manager	outside scope of system
Nurse	class

system	outside scope of system
username	Redundant
password	Redundant
appointments	Redundant
assistant	Redundant
health report	Class(boundary)
hospital	outside scope of system
blood orders	class
OrderNO	attribute
OrderDate	attribute
blood requirement	outside scope of system

Classes

- Donor
- Appointment
- Campaign
- Assistant
- Order
- Nurse
- Budget Report
- Health Report

CRC Cards

Class Name: Donor	
Responsibilities	**Collaborations**
Register to the system	
Submit personal details	
Login to the system	
Update personal details	
Make and cancel appointments	Appointment

Class Name: Assistant	
Responsibilities	**Collaborations**
Login to the system	
Organize campaigns	Campaign
Make budget reports	Budget Report
Manage orders	Order

Class Name: Nurse	
Responsibilities	**Collaborations**
Login to the system	
Check appointments	Appointment
Update health reports	Health Reports

Class Name: Order	
Responsibilities	**Collaborations**
Get order list	

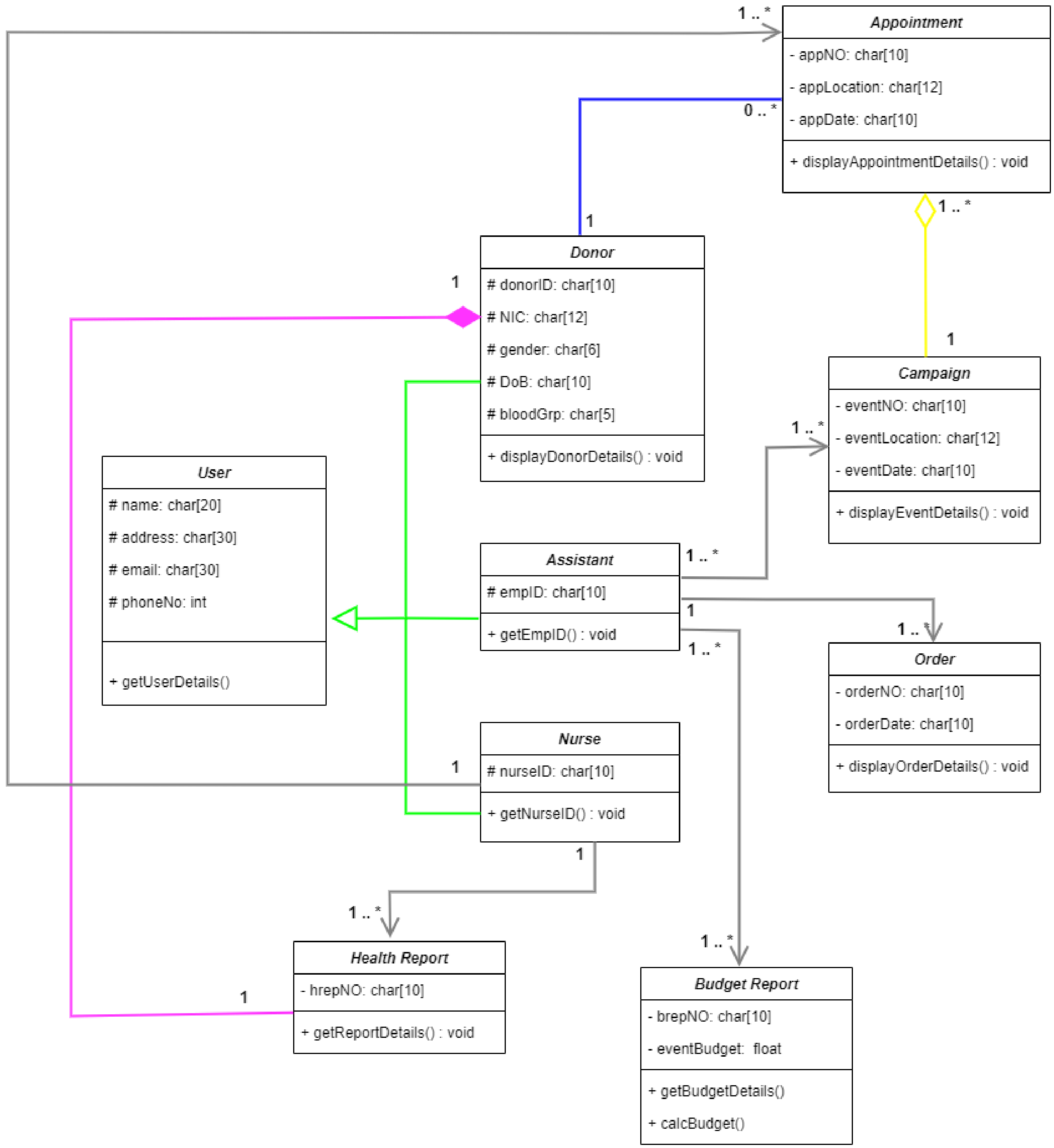
Class Name: Campaign	
Responsibilities	**Collaborations**
Set campaign details	

Class Name: Appointment	
Responsibilities	**Collaborations**
Set appointment details	Donor, Campaign

Class Name: Budget Report	
Responsibilities	**Collaborations**
Calculate Annual Budget	

Class Name: Health Report	
Responsibilities	**Collaborations**
Update health reports	Donor
Submit health reports	Assistant

Class Diagram



C++ Coding

User.h

```
class User {
  protected:
    char name[20];
    char address[30];
    char email[30];
    int phoneNo;
  public:
    User();
    User(char pName[], char pAddress[], char pEmail[], int pPhone);
    void getUserDetails();
};
```

User.cpp

```
#include <iostream>
#include <string>
#include "User.h"
using namespace std;

User::User(){
  name = "";
  address = "";
  email = "";
  phoneNo = 0;
}
User::User(string pName, string pAddress, string pEmail, int pPhone){
  name = pName;
  address = pAddress;
  email = pEmail;
  phoneNo = pPhone;
}
void User::getUserDetails(){
  cout << "Enter Name : " << name << endl;
  cout << "Enter Address : " << address << endl;
  cout << "Enter Email : " << email << endl;
  cout << "Enter Phone No : " << phoneNo << endl;
}
```

Assistant.h

```
#pragma once
#include "User.h"
#include "campaign.h"
#include "BudgetReport.h"
#include "Order.h"

class Assistant : public User {
  protected:
    string empID;
    campaign *camp;
    Order *order;
    BudgetReport *budget;
  public:
    Assistant();
    Assistant(campaign *c, Order *or, BudgetReport *b, string pName, string pAddress, string pEmail, int pPhone, string pEID) : User (pName, pAddress, pEmail, pPhone);
    void displayEmpID();
    ~Assistant();
};
```

Assistant.cpp

```
#include <iostream>
#include <string>
#include "Assistant.h"
using namespace std;

Assistant::Assistant(){};

Assistant::Assistant(campaign *c, Order *or, BudgetReport *b, string pName, string pAddress, string pEmail, int pPhone, string pEID) : User (pName, pAddress, pEmail, pPhone){
    empID = pEID;
    camp = c;
    order = or;
    budget = b;
}
```

```
void Assistant::displayEmpID(){
    cout << "Assistant ID : " << empID << endl;
    camp -> displayEventDetails();
    order -> displayOrderDetails();
    budget -> displayDetails();
    budget -> calcBudget();
}

Assistant::~Assistant() {};
```

Donor.h

```
#pragma once
#include <string>
#include "HealthReport.h"
#include "User.h"

#define SIZE 2

using namespace std;

class Donor:public User{
  protected:
   string donorID;
   string NIC;
   string gender;
   string DoB;
   string bloodGrp;
   HealthReport *reprt[SIZE];
   Appointment *order[SIZE];
                    int noOfAppointments;

  public:
    Donor();
    Donor(string pName, string pAddress, string pEmail, string pPhone,string did, string nic, string gndr,
string DOB, string blgrp,string no1, string no2);
      void addAppointment(Appointment *A);
                    void displayDonorDetails();
    ~Donor();
};
```

Donor.cpp

```
#include <iostream>
#include <string>
#include "Donor.h"
#include "HealthReport.h"
#define SIZE 2
int i=0;
using namespace std;

Donor::Donor()
{
     donorID = "";
     NIC = "";
     gender = "";
     DoB = "";
     bloodGrp = "";
     reprt[0] = new HealthReport(0);
     reprt[1] = new HealthReport(0);
    }
Donor::Donor(string pName, string pAddress, string pEmail, string pPhone,string did, string nic, string gndr, string DOB, string blgrp,string no1, string no2):User(pName, pAddress, pEmail, pPhone)
{
       donorID = did;
       NIC = nic;
       gender = gndr;
       DoB = DOB;
       bloodGrp = blgrp;
       noOfAppointments = 0;
       reprt[0] = new HealthReport(no1);
       reprt[1] = new HealthReport(no2);
    }
void Donor::DisplayDonorDetails() {
       cout<<"DonorID : "<<donorID<<endl;
       cout<<"NIC : "<<NIC<<endl;
       cout<<"Gender : "<<gender<<endl;
       cout<<"Date of Birth : "<<DoB<<endl;
       cout<<"Blood Group : "<<bloodGrp<<endl;

       for (int r=0; r<SIZE; r++){
         reprt[r]->getreportdetails();
       }
```

```
        for( int i = 0; i < noOfAppointments; i++){
          order[i]-> displayOrders();
        }
}
void Donor::addAppointment(Appointment *A)
{
        if (noOfAppointments < SIZE)
           order[noOfAppointments] = A;
        noOfAppointments++;
}
Donor::~Donor() {
        cout << "******************************"<< endl;
        cout << "Donor Removing " << endl;
        cout<<endl;
        for (int r=0; r<SIZE; r++){
           delete reprt[r];
          }
}
```

Nurse.h

```
#pragma once
#include "User.h"
#include "Healthreport.h"
#include "appointment.h"


class Nurse : public User {
  private:
    string nurseID;
    Healthreport *rpt;
    appointment *appoint;
  public:
    Nurse();
    Nurse(appointment *ap, Healthreport *prpt, string pName, string pAddress, string pEmail, int pPhone,
string pNID) : User (pName, pAddress, pEmail, pPhone);
    void getNurseID();
    ~Nurse();
};
```

Nurse.cpp

```
#include <iostream>
#include <string>
#include "Nurse.h"
using namespace std;

Nurse::Nurse() {};

Nurse::Nurse(appointment *ap, Healthreport *h, string pName, string pAddress, string pEmail, int
pPhone, string pNID) : User (pName, pAddress, pEmail, pPhone) {
  nurseID = pNID;
  appoint = ap;
  health = h;
}

void Nurse::getNurseID() {
  cout << "Nurse ID : " << nurseID << endl;
  appoint -> displayAppointmentDetails();
  health -> getreportdetails();
}

Nurse::~Nurse() {};
```

Appointment.h

```
#pragma once
#include <string>
#include "Donor.h"
#include "campaign.h"

using namespace std;

class Donor;
class campaign;

class Appointment
{
  private:
        string appNO;
        Donor *Don;
        string appLocation;
        string appDate;

   public:
          Appointment(string appointmentNO, string appointmentLO, string appointmentDate,Donor
*Don);
        void displayAppointmentDetails();
        ~Appointment();
};
```

Appointment.cpp

```
#include <iostream>
#include <string>
#include "Appointment.h"
#include "Donor.h"

using namespace std;

Appointment::Appointment(string appointmentNO, string appointmentLO, string
appointmentDate,Donor *Don){
        appNO=appointmentNO;
        appLocation=appointmentLO;
        appDate=appointmentDate;
        Don = pDon;
        Don->addAppointment(this);
   }
void Appointment::displayAppointmentDetails(){
        cout << "Appointment NO :" << appNO << endl;
        cout << "Appointment Location :" << appLocation << endl;
        cout << "Appointment Date :" << appDate << endl;
  }
Appointment::~Appointment(){
        cout << "Deleting Appointment" << appNO << endl;
}
```

BudgetReport.h

```
#pragma once
#include <string>
#include "Assistant.h"

using namespace std;

class Assistant;

class BudgetReport
{
        private:
                string budgtReportID;

        public:
                BudgetReport();
                BudgetReport(string bgreportID);
                void displayBudgetReport();

};
```

BudgetReport.cpp

```
#pragma once
#include <string>
#include "BudgetReport.h"

using namespace std;

BudgetReport::BudgetReport(){}
BudgetReport::BudgetReport(string bgreportID)
{
	budgtReportID = bgreportID ;
}
void BudgetReport::displayBudgetReport()
{
	cout << " Budget ReportID  = " << budgtReportID << endl;
}
```

campaign.h

```
#pragma once
#include <iostream>
#include <cstring>
class Appointment;
class campaign {
  private:
    char eventNO[10];
    char eventLocation[12];
    char eventDate[10];
    appointment *app[SIZE];

  public:
    campaign();
    void displayEventDetails();
    ~campaign();
};
```

campaign.cpp

```
#include <iostream>
#include <string>
#include "campaign.h"

campaign::campaign(){}
campaign::void addappointment(appointment *app1, appointment *app2)
                {
                        app[0] = app1;
                        app[1] = app2;
                }

void campaign::displayEventDetails(){
    cout<<"EventNo="<< eventNO <<endl;
    cout<<"Event Location="<< eventLocation <<endl;
    cout<<"Event Date="<< eventDate <<endl;

    for(int i = 0; i < SIZE; i++)
                app[i]->displayAppointmentDetails();
   }

campaign::~campaign(){
 cout << "Deleting campaign" << endl;
}
```

HealthReport.h

```
#pragma once
#include <iostream>
#include <string>
class HealthReport {
  private:
    char hrepNO;

  public:
    HealthReport();
    HealthReport(string hrNO);
    void getreportdetails();
    ~HealthReport();
};
```

HealthReport.cpp

```
#include <iostream>
#include <string>
#include "HealthReport.h"
using namespace std;

HealthReport::HealthReport(){
    hrepNO = "";
       }

HealthReport::HealthReport(string hrNO){
    hrepNO = hrNO;
       }

void HealthReport::getreportdetails(){
              cout <<  " health report: "  << hrepNO << endl;
              }

HealthReport::~HealthReport(){};
```

Order.h

```
#pragma once

class Order{
private:
  string orderNO;
  string orderDate;

public:
  Order();
  Order(string Onumber,string Odate);
  void dispalyOrderDetais();
  ~Order();
};
```

Order.cpp

```
#include "Order.h"
#include <iostream>
#include <string>
using namespace std;

Order::Order(){};

Order::Order(string Onumber,string Odate){
        orderNo= Onumber;
        orderDate=Odate;
    }
void Order :: dispalyOrderDetais(){
        cout << "Order No : " << orderNo << endl;
        cout << "Order Date : " << orderDate << endl;
    }
Order::~Order(){};
```

Main.cpp

```
#include <iostream>
#include <string>
#include "Appointment.h"
#include "Assistant.h"
#include "BudgetReport.h"
#include "campaign.h"
#include "Donor.h"
#include "HealthReport.h"
#include "Nurse.h"
#include "Order.h"
#include "User.h"

using namespace std;

int main() {

  Appointment *app = new Appointment("A001" , "Colombo" , "05-06-2022");
  app -> displayAppointmentDetails();
  Assistant *assi = new Assistant("001" , "O001" , "BR001" , "Nimali" , "Nugegoda" ,
"nimali69@yahoo.com" , "0712965678" , "Em002");
  assi -> displayEmpID();

  BudgetReport *br = new BudgetReport("BR001");
  br -> displayBudgetReport();

  campaign *c = new campaign ("001" , "Piliyandala" , "06-10-2022");
  c -> displayEventDetails();

  Donor *d = new Donor("Kasun" , "Malabe" , "kasun@gmail.com" , "0777771235" , "D002" ,
"695236841V" , "male" , "11-02-1969" , "A+" , "H001" , "H005");
  d -> DisplayDonorDetails();

  HealthReport *hr = new HealthReport("H001");
  hr -> getreportdetails();

  Nurse *n = new Nurse("A001" ,"H001" , "Kamani" , "Badulla" , "kamani@yahoo.com" , "0715236982" ,
"N005");
  n -> getNurseID();

  Order *or = new Order("O001" , "20-06-2022");
  or -> dispalyOrderDetais();

  return 0;
}
```