Topic : Blood Donation System

Group no : MLB_WD_CSNE_13_08

Campus : Malabe

Submission Date : 17/05/2022

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**
IT21083532	Yatawarage U. S	071 2463380
IT21339042	Hettiarachchi L. R	077 5789117
IT21928192	Kavindya J. A. G	071 5890795
IT21383434	Muhandiramge M. D. A. D	077 6733940
IT21108372	Karunarathna U. G. O. S	070 2062585

Content

System Requirements

- System can access 24/7
- Donator can donate his blood to any Bloodcamp.
- Donator should gives his details. There are, donarId, fName, lName, Address, Pno, Email, BloodType, DoB
- Bloodcamp store the donator's details.
- Every Bloodcamp has one or more Medicalperson.
- Medicalperson should take the blood and all the Bloodcamp information.
- Medicalperson should update the Bloodbank system.
- Users can channel a doctor and take their requirements.
- Doctors can check blood availability.
- Bloodbank transfer the blood to the Users.

Noun & Verb Analysis

-Nouns-

- **System** can access 24/7.
- **Donator** can donate **his** **blood** to any **Bloodcamp**.
- **Donator** should gives **his** **details**. There are, **donarId**, **fName**, **lName**, **Address**, **Pno**, **Email**, **BloodType**, **DoB**.
- **Bloodcamp** store the donator's **details**.
- Every **Bloodcamp** has one or more **Medicalperson**.
- **Medicalperson** should take the **blood** and all the **Bloodcamp** **information**.
- **Medicalperson** should update the **Bloodbank** **system**.
- **Users** can channel a **doctor** and take their requirements.
- **Doctors** can check **blood** availability.
- **Bloodbank** transfer the **blood** to the **Users**.

Noun & Verb Analysis

-Verbs-

- System can **access** 24/7
- Donator can **donate** his blood to any Bloodcamp.
- Donator should **gives** his details. There are, donarId, fName, lName, Address, Pno, Email, BloodType, DoB
- Bloodcamp **store** the donator's details.
- Every Bloodcamp has one or more Medicalperson.
- Medicalperson should **take** the blood and all the Bloodcamp information.
- Medicalperson should **update** the Bloodbank system.
- Users can **channel** a doctor and **take** their requirements.
- Doctors can **check** blood availability.
- Bloodbank **transfer** the blood to the Users.

Identified Class

- ✓ Donator
- ✓ Bloodcamp
- ✓ Medicalperson
- ✓ Bloodbank
- ✓ User
- ✓ Doctor

Reason for rejecting other nouns

(01)Redundant

- ➢ His refers to the same person as “Donator”.

(02)Outside scope of system

- ➢ Blood and System is an outside scope of system.

(03)An attribute

- ➢ donarId , fName, lName, Address, Email, Pno , BloodType, DoB those are an attributes in “Donator”

CRC Cards

_D

Class Name: Donator	
Responsibilities	**Collaboration**
Offer to donate blood	

Class Name: Bloodbank	
Responsibilities	**Collaboration**
Check Aailability	Doctor
Release of blood	Doctor

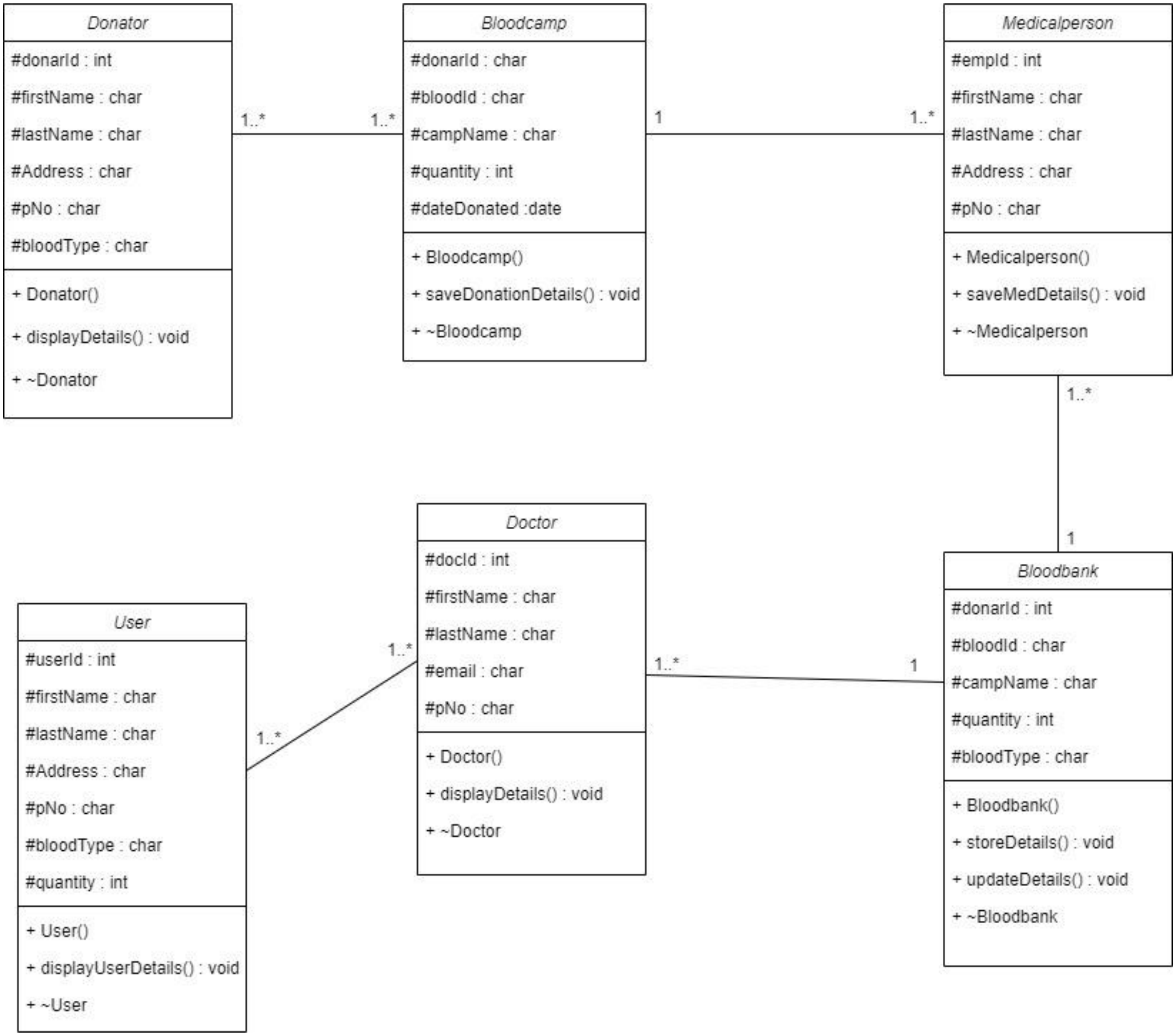
Class Name: Medicalperson	
Responsibilities	**Collaboration**
Collect Blood	
Organizing Blood Camp	

Class Name: User	
Responsibilities	**Collaboration**
Applying for blood transfusion	Doctor

Class Name: Doctor	
Responsibilities	**Collaboration**
Check Aailability	Bloodbank
Get user requests	User

Class Name: Bloodcamp	
Responsibilities	**Collaboration**
Medical check up	
Register Details	
Blood Transfusion	

Class Diagram (UML Notation)



Class Headers

Donator.h

```
#include<iostream>

using namespace std;

class Donator
{
protected:
       int donarId;
       char firstName[20];
       char lastName[20];
       char Address[30];
       char pNo[10];
       char bloodType[3];

public:
       Donator();
       Donator(int pdonarId, const char pfirstName[], const char plastName[],
const char pAddress[], const char ppNo[], const char pbloodType[]);
       void displayDetails();
       ~Donator();

};
```

Bloocamp.h

```
#include<iostream>

using namespace std;

class Bloodcamp
{
protected:
       int donarId;
       char bloodId;
       char campName[20];
       int quantity;
       char dateDonated;

public:
       Bloodcamp();
       Bloodcamp(int pdonarId, const char pbloodId, const char pcampName[], int
pquantity, const char pdateDonated);
       void saveDonationDetails();
       ~Bloodcamp();
};
```

Medicalperson.h

```
#include<iostream>

using namespace std;

class Medicalperson
{
protected:
       int empId;
       char firstName [20];
       char lastName[20];
       char Address [30];
       char pNo [10];

public:
       Medicalperson();
       Medicalperson(int pempId, const char pfirstName[], const char plastName[],
const char pAddress[], const char ppNo[]);
       void saveMedDetails();
       ~Medicalperson();
};
```

Bloodbank.h

```
#include<iostream>

using namespace std;
class Bloodbank
{
protected:
       int donarId;
       char bloodId;
       char campName[20];
       int quantity;
       char bloodType[4];

public:
       Bloodbank();
       Bloodbank(int pdonarId, const char pbloodId, const char pcampName[], int
pquantity , const char pbloodType[]);
       void updateDetails();
       void storeDetails();
       ~Bloodbank();
};
```

Doctor.h

```
#include<iostream>

using namespace std;
class Doctor
{
protected:
        int docId;
        char firstName[20];
        char lastName[20];
        char email[50];
        char pNo [10];

public:
        Doctor();
        Doctor(int pdocId, const char pfirstName[], const char plastName[], const 
char pemail[], const char ppNo[]);
        void displayDetails();
        ~Doctor();
};
```

User.h

```
#include<iostream>

using namespace std;
class User
{
protected:
        int userId;
        char firstName[20];
        char lastName[20];
        char Address[30];
        char pNo[10];
        char bloodType[3];
        int quantity;

public:
        User();
        User(int puserId, const char pfirstName[], const char plastName[], const 
char pAddress[], const char ppNo[], const char pbloodType[]), int pquantity);
        void displayUserDetails();
        ~User();
};
```

Donator.cpp

```
#include"Donator.h"
#include<cstring>


Donator::Donator()
{
	donarId = 0;
	strcpy(firstName, "");
	strcpy(lastName, "");
	strcpy(Address, "");
	strcpy(pNo, "000000000");
	strcpy(bloodType, "");
}
Donator::Donator(int pdonarId, const char pfirstName[], const char
plastName[], const char pAddress[], const char phNo[], const char
pbloodType[])
{
	donarId = pdonarId;
	strcpy(firstName, pfirstName);
	strcpy(lastName, plastName);
	strcpy(Address, pAddress);
	strcpy(pNo, phNo);
	strcpy(bloodType, pbloodType);
}

void Donator::displayDetails()
{
}
Donator::~Donator()
{
	//Destructor
}
```

Bloodcamp.cpp

```
#include"Bloodcamp.h"
#include<cstring>
#include "Donator.h"

Bloodcamp::Bloodcamp()
{
	donarId = 0;
	strcpy_s(bloodId, "");
	strcpy(campName, "");
	quantity = 0;
	strcpy(dateDonated, "");
}
Bloodcamp :: Bloodcamp(int pdonarId, const char pbloodId[], const char
pcampName[], int
```

```
	pquantity, const char pdateDonated[])
{
	donarId = pdonarId;
	strcpy(bloodId,pbloodId);
	strcpy(campName, pcampName);
	quantity = pquantity;
	strcpy(dateDonated, pdateDonated);

}void Bloodcamp ::saveDonationDetails()
{
}
Bloodcamp::~Bloodcamp()
{
	//Destructor
}
```

Medicalperson.cpp

```
#include"Medicalperson.h"
#include<cstring>


Medicalperson::Medicalperson()
{
	empId = 0;
	strcpy(firstName, "");
	strcpy(lastName, "");
	strcpy(Address, "");
	strcpy(pNo, "000000000");

}
Medicalperson::Medicalperson(int pempId, const char pfirstName[], const char
plastName[],
	const char pAddress[], const char pNo[])
{
	empId = pempId;
	strcpy(firstName, pfirstName);
	strcpy(lastName, plastName);
	strcpy(Address, pAddress);
	strcpy(pNo, pNo);

}

void Medicalperson::saveMedDetails()
{

}
Medicalperson::~Medicalperson()
{
	//Destructor
}
```

Bloodbank.cpp

```
#include"Bloodbank.h"
#include"Donator.h"
#include<cstring>

Bloodbank::Bloodbank()
{
        donarId = 0;
        strcpy_s(bloodId, "");
        strcpy_s(campName, "");
        quantity = 0;
        strcpy_s(bloodType, "");

}
Bloodbank::Bloodbank(int pdonarId, const char pbloodId, const char
pcampName[], int
        pquantity, const char pbloodType[])

{
        donarId = pdonarId;
        strcpy(bloodId, pbloodId);
        strcpy(campName, pcampName);
        strcpy(bloodType, pbloodType);
        quantity = pquantity;
}

void Bloodbank::updateDetails()
{

}
void Bloodbank::storeDetails()
{

}
 Bloodbank ::~Bloodbank()
{

}
```

Doctor.cpp

```
#include"Doctor.h"
#include<cstring>

Doctor::Doctor()
{
        docId = 0;
        strcpy(firstName, "");
        strcpy(lastName, "");
        strcpy(email, "");
        strcpy(pNo, "");
}

Doctor::Doctor(int pdocId, const char pfirstName[], const char plastName[],
const
        char pemail[], const char ppNo[])
{
        docId = pdocId;
        strcpy(firstName, pfirstName);
        strcpy(lastName, plastName);
        strcpy(email, pemail);
        strcpy(pNo, ppNo);
}


void Doctor::displayDetails()
{

}
Doctor::~Doctor()
{

}
```

User.cpp

```
#include"User.h"
#include<cstring>

User::User()
{
	userId = 0;
	strcpy(firstName, "");
	strcpy(lastName, "");
	strcpy(Address, "");
	strcpy(pNo, "");
	strcpy(bloodType, "");

	quantity = 0;
}

User::User(int puserId, const char pfirstName[], const char plastName[], 
const
	char pAddress[], const char ppNo[], const char pbloodType[], int 
pquantity)
{
	userId = puserId;
	strcpy(firstName, pfirstName);
	strcpy(lastName, plastName);
	strcpy(Address, pAddress);
	strcpy(pNo, ppNo);
	strcpy(bloodType, pbloodType);
	quantity = pquantity;
}

void User::displayUserDetails()
{

}
User ::~User()
{

}
```

Main.cpp

```cpp
#include"Donator.h"
#include"Bloodcamp.h"
#include "Doctor.h"
#include"Medicalperson.h"
#include"Bloodbank.h"
#include"User.h"
#include<iostream>

using namespace std;

int main()
{

	Donator* d1 = new Donator(0001, "Kasun", "Kalhara","No 12, Samagi
mawatha,  Panniptitiya", "0702064535", "B+");
	Donator* d2 = new Donator(0002, "Saman", "Shantha","No 12, Salmal
mawatha,  Nittabuwa", "0702034545", "O+");

	Bloodcamp *camp1 = new Bloodcamp (0001, "B001", "APEKSHA", 300,
"04/25/2022");
	Bloodcamp* camp2 = new Bloodcamp(0002, "B002", "SUMITURA", 400,
"04/05/2022");

	Bloodbank *bank1 = new Bloodbank(0001, "B001", "APEKSHA", 300, "B+ ");
	Bloodbank *bank2 = new Bloodbank(0002, "B002", "SUMITURA", 400, "O+ ");

	Doctor *doc1 = new Doctor(2001, "Pradeep", "Rangana",
"pradeep@gmail.com", "0702062585");
	Doctor* doc2 = new Doctor(2050, "Amal", "Perera", "amal@gmail.com",
"0702062875");

	Medicalperson *med1 =new Medicalperson(11001, "Nuwan", "Jayasignha",
"No 23, Hospital Road, Ratnapura", "0702733872");
	Medicalperson* med2 = new Medicalperson(11003, "Dilina", "Guruge", "No
43, Malwatta Road, Kandy", "0702839870");

	User *user1 = new User(12001, "Oshan", "Amantha", "No32, Wihara
mawatha, Pannala", "07288774567", "O+", 300);
	User* user2 = new User(12001, "Dulith", "Gamage", "No43, School lane,
Ja-ela", "0725476667", "O+", 400);
	return 0;
}
```