# Image result for nust logo

# *Department of Computing*

# *CS220: Database Management Systems*

# *Instructor: Ms. Hirra Anwar*

# *Class: BSCS-7A*

# *Project Deliverable 3:Project Report.*

# *Date: 21th Dec 2018*

***Submitted by:***

***Muhammad Nouman Imran (215192)***

***Hasnat Amir (210750)***

***Faizan Abid Awan (120452)***

Data Base System **End** Semester Project

* Blood donation system

# Explanation:

**Description:**

1-priorities for the patients depending upon their medical conditions

2-blood balance (how much is coming and how much is going out)

3-time limitation for the donor (cannot give bold for 3 months after donation)

4-checking the medical history and current situation of the donors

5-check and balance with other blood banks around the area for mutual cooperation of supplies of blood in case of shortage.

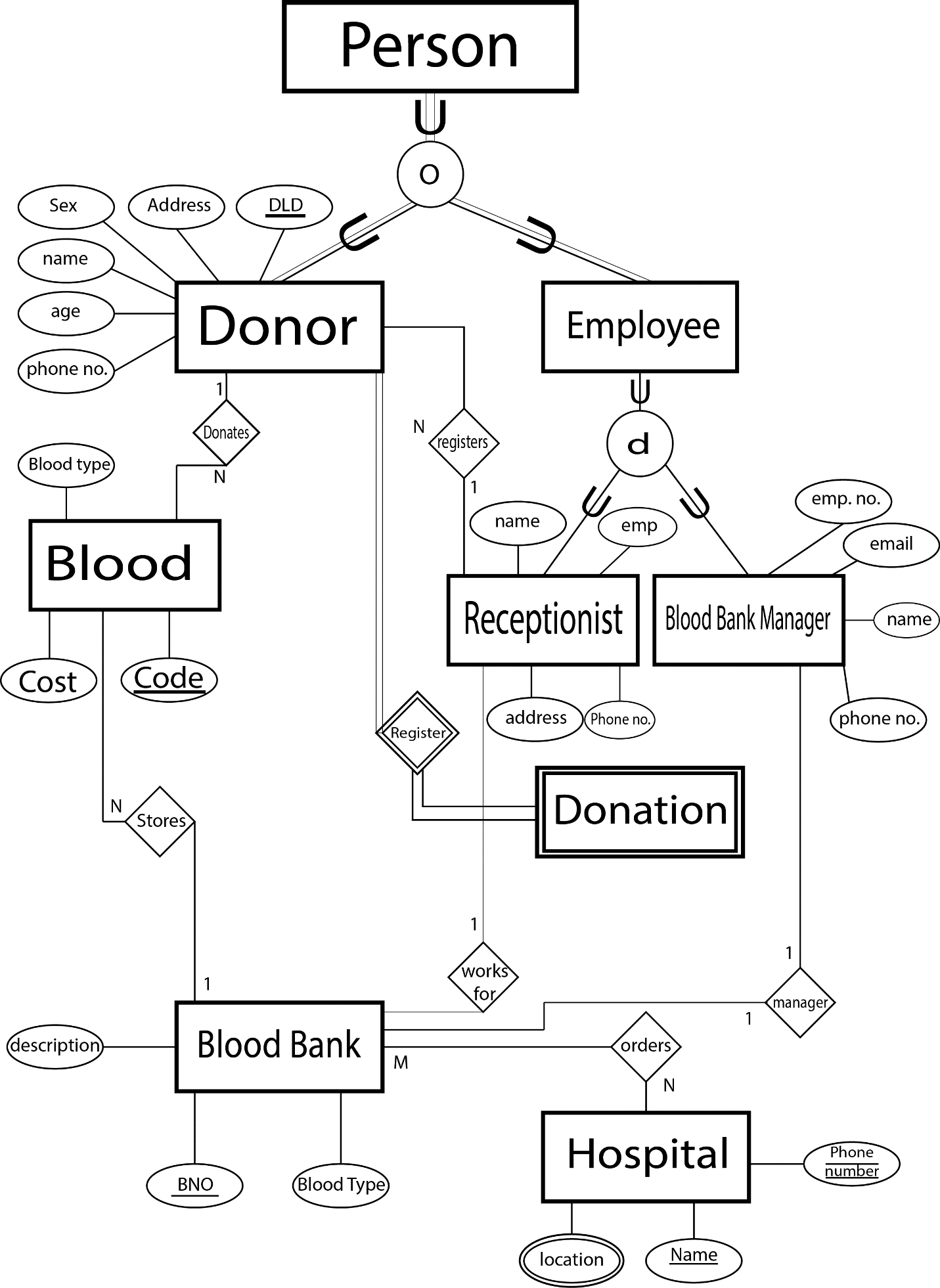
**Key schemas:**

Data of patient: patient name, patient id, blood type, BMI, patient medical condition,

Data of donor: name, ID, blood type, BMI, last donated (date), medical history, address, contact number

Data of blood bank: blood bank name, bank address, donor name, bank contact number, etc.

**ERD:**



The person class consists of all attributes that a person can have. There are two SUBTYPES of person which are Donor and Employee. Donor is uniquely identified by donor id and EMPLOYEE BY Employee id. The employee is further classified into Receptionist or a Blood Bank Manager. A donor must registers for the Donation he made. Hospital can have multiple locations and each hospital can order blood from blood bank. Hospital Name is the primary key with phone no as well which will make it composite key. A blood is stored in a blood bank. One blood bank manager manages only one Blood bank. A Receptionist works for blood bank. The blood is given a unique code to identify it uniquely. A receptionist registers itself for the Donors. Each blood bank has a unique bank number and description about how its formed and what services it offers. An employee can also be a donor shown by overlapping constraints but an employee can be only a receptionist or a Manager. Donation is a weak entity type and every donor must registers before the donation.

**Mappings:**

# Person (SSN, address, Phone number, Sex )

Donor (Did, SSN ,name , age , Code )

Employee(Eid, SSN ,name ,email)

Blood(Code , cost , Blood type)

Blood Bank(BNo , blood type, description, code)

Hospital( License Number, Name , Phno)

Hospital Location (License Number , Hospital-location)

Receptionist(Eid , address , phno , working hours, date joined, salary , BNo, Did)

Blood bank Manager(Eid, phno , date joined , working hours , salary ,BNo)

Donation( Did, Date-Of-Doantion , Blood-Group-Donated)

Orders(License Number,BNo)

**\*Arrows for foreign key/primary key relationship are present on the hard copy submitted in the lab.**

**Functional Dependencies:**

**In Person table:**

SSN --------------------> address, sex, phone no

SSN ,ph no--------------> address ,sex

**In Donor table:**

Did---------------------> name , age ,code

**In Employee table:**

Eid----------------------> name , email

**In blood table:**

Code-------------------> cost , blood type

**In blood bank table:**

Bno----------------------> blood type , description , code

**In Hospital table:**

License Number------------> Name , phone\_no

**In Hospital Location table:**

License Number---------> Hospital location

**In Receptionist table:**

Eid ----------------> working hours , salary , ph no , address , date joined, Bno, Did

Ph no--------------> working hours , salary , address , date joined

**In Blood bank Manager:**

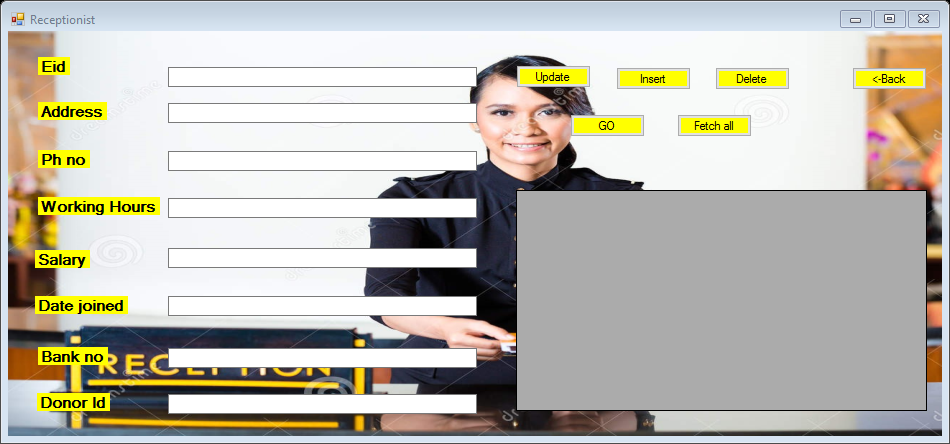
Eid------------------> phno , date joined , working hours , salary, Bno

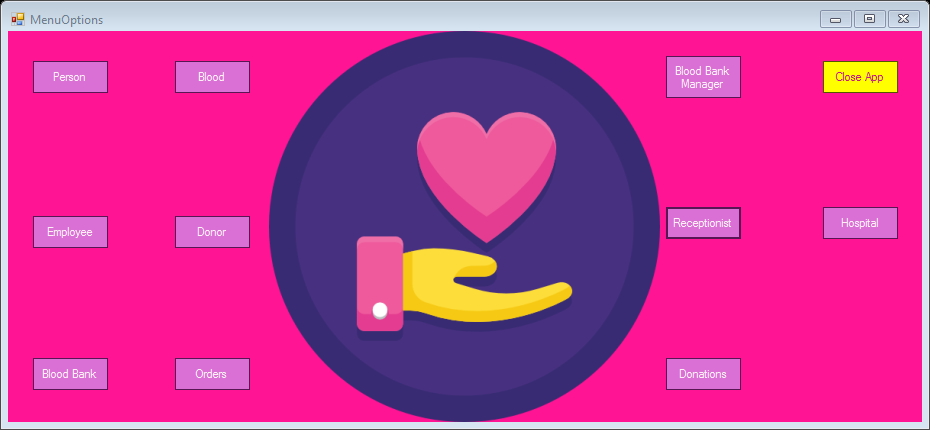
Phno ---------------> working hours , salary , date joined

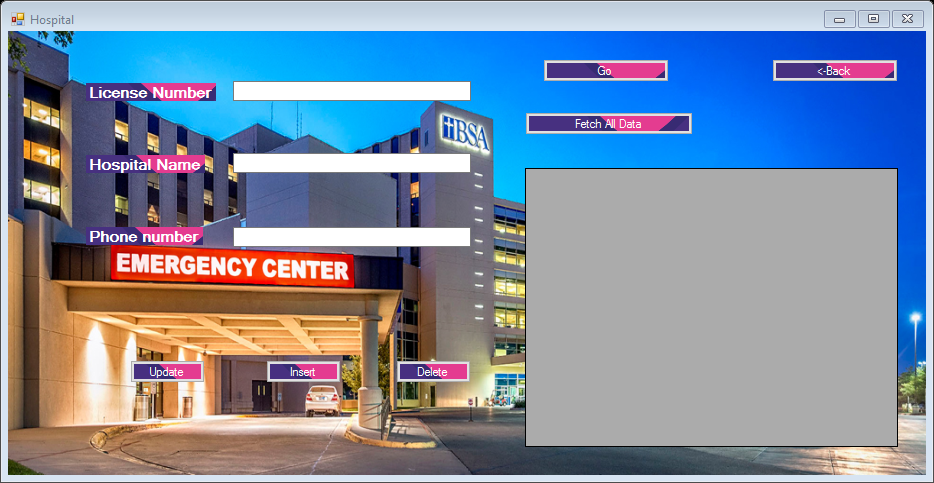
**In Donation table:**

Did -----------------> Date-Of-Doantion , Blood-Group-Donated

**Frontend:**







The frontend is made in Visual C++ with delete, insert, update and select and select all functionality.

**Connection with mysQl:**

//insert data even

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

//Insert Onclick Handler

//MessageBox::Show("Insert clicked!!!");

try{

String^ info = "Server=127.0.0.1;Uid=root;Pwd=;Database=blood bank management system";

MySqlConnection^ con = gcnew MySqlConnection(info);

int Did = Int32::Parse(textBox1->Text);

int SSN = Int32::Parse(textBox5->Text);

String^ Name = textBox4->Text;

String^ Age = textBox3->Text;

int Code = Int32::Parse(textBox2->Text);

MySqlCommand^ cmd = gcnew MySqlCommand("insert into donor values(" + Did + "," + SSN + ",'" + Name + "','" + Age + "'," + Code + ")", con);

con->Open();

MySqlDataReader^ dr;

dr = cmd->ExecuteReader();

MessageBox::Show("User info saved!!!!");

con->Close();

}

catch (Exception^ ex){

MessageBox::Show(ex->Message);

}

}

private: System::Void textBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

}

//go button event

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

try{

int Did = Int32::Parse(textBox1->Text);

String^ info = "Server=127.0.0.1;Uid=root;Pwd=;Database=blood bank management system";

MySqlConnection^ con = gcnew MySqlConnection(info);

MySqlCommand^ cmd = gcnew MySqlCommand("Select \* from donor where Did=" + Did + "", con);

con->Open();

MySqlDataReader^ dr;

dr = cmd->ExecuteReader();

while (dr->Read()){

textBox5->Text = dr->GetString(1);

textBox4->Text = dr->GetString(2);

textBox3->Text = dr->GetString(3);

textBox2->Text = dr->GetString(4);

}

con->Close();

}

catch (Exception^ ex){

MessageBox::Show(ex->Message);

}

}

//fetch all data event

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

try{

String^ info = "Server=127.0.0.1;Uid=root;Pwd=;Database=blood bank management system";

MySqlConnection^ con = gcnew MySqlConnection(info);

MySqlDataAdapter^ sda = gcnew MySqlDataAdapter("select \* from donor", con);

DataTable^ dt = gcnew DataTable();

sda->Fill(dt);

bindingSource1->DataSource = dt;

dataGridView1->DataSource = bindingSource1;

}

catch (Exception^ ex){

MessageBox::Show(ex->Message);

}

}

private: System::Void form1\_Load(System::Object^ sender, System::EventArgs^ e) {

}

//update event

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

try{

String^ info = "Server=127.0.0.1;Uid=root;Pwd=;Database=blood bank management system";

MySqlConnection^ con = gcnew MySqlConnection(info);

int Did = Int32::Parse(textBox1->Text);

int SSN = Int32::Parse(textBox5->Text);

String^ Name = textBox4->Text;

String^ Age = textBox3->Text;

int Code = Int32::Parse(textBox2->Text);

MySqlCommand^ cmd = gcnew MySqlCommand("Update donor set Name='" + Name + "',Age='" + Age + "',Code=" + Code +" where Did="+Did+" and SSN="+SSN+"", con);

con->Open();

MySqlDataReader^ dr;

dr = cmd->ExecuteReader();

MessageBox::Show("User info updated!!!!");

con->Close();

}

catch (Exception^ ex){

MessageBox::Show(ex->Message);

}

}

//delete event handler

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

try{

String^ info = "Server=127.0.0.1;Uid=root;Pwd=;Database=blood bank management system";

MySqlConnection^ con = gcnew MySqlConnection(info);

int Did = Int32::Parse(textBox1->Text);

int SSN = Int32::Parse(textBox5->Text);

MySqlCommand^ cmd = gcnew MySqlCommand("delete from donor where Did="+Did+" AND SSN="+SSN+"",con);

con->Open();

MySqlDataReader^ dr;

dr = cmd->ExecuteReader();

MessageBox::Show("User info deleted!!!!");

con->Close();

}

catch (Exception^ ex){

MessageBox::Show(ex->Message);

}