Karen Mani & Sasha Smolyansky

Professor He

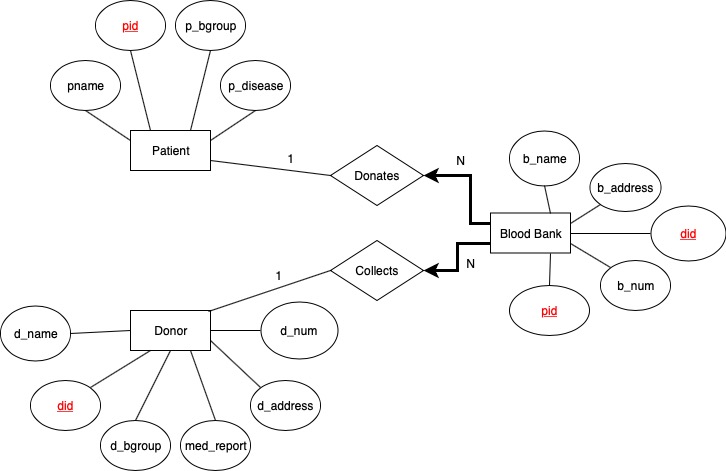
CSCI 2441W - Project 2

27 April 2019

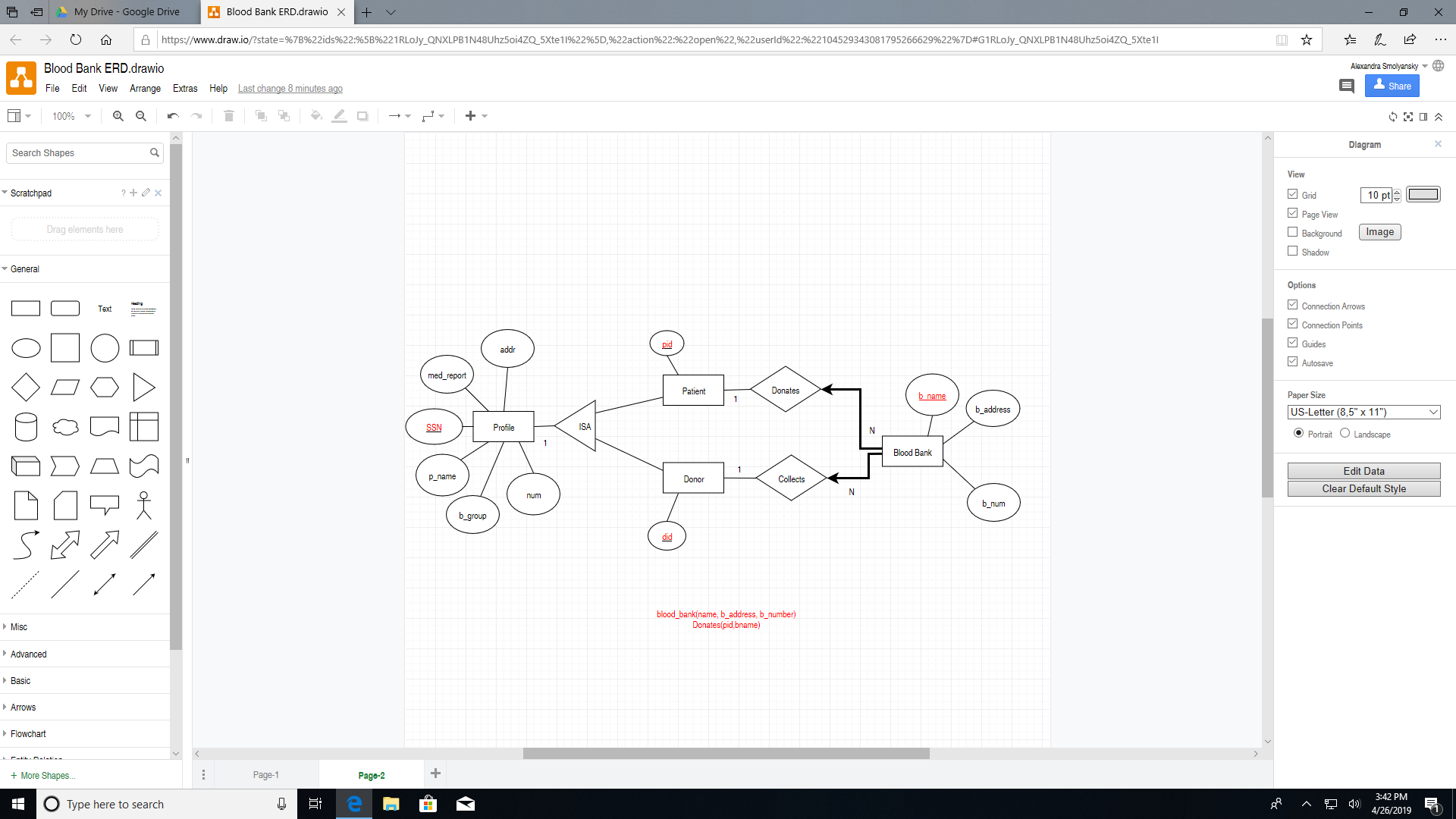
Project 2 Report

**1.1 SCHEMA REFINEMENT AND NORMAL FORMS**

**1.1.1 Project 1 ERD**

****

**1.1.2 Refined ERD**

****

In our refined ERD, we decided to change the format of our database a bit. We altered the attributes of most of our tables so that Patients and Donors are only defined by their ID’s. Profile is now connected to Patients and Donors with an ISA, which means that Profile is the parent function and Patient/Donor are the children of Profile. Essentially, Profile now holds data regarding all the information about the patient and/or donor like SSN, name, blood type, phone number, address, and any medical history. Because of the ISA function, patient and donor assume those attributes as well. Blood Banks are now characterized by their name, address, and phone number in order to uniquely identify patients in a more concise manner than we were doing so. Connecting the Blood Bank to pids and dids as we did before was redundant and unnecessary if the Blood Bank is also holding profiles that contained all that information for the Patients and Donors itself. Patients and the Blood Bank have a ‘donates’ relationship because blood banks donate to patients in a 1-N relationship (meaning blood banks can donate to many patients but the patient can only receive blood from one bank). Similarly, blood banks can ‘collect’ blood from donors in the same sense that they can collect from multiple donors but donors can only donate their blood to one bank.

Through this process, we learned the importance of redundancy, as this can create errors when displaying data. We learned that because we were defining so many of our tables by one attribute. When we tried to perform certain functions on our data, we would get repeats of the same patient or donor in the output. Additionally, we learned how to use the ISA function and how a child can embody all the attributes of a parent entity.

**1.1.3 Identify Functional Dependencies**

**1.1.3.1 Updated Schema:**

Patients (*pid*: integer)

Donors (*did*: integer)

Blood Blank (*b\_name*: string, *b\_address*: string, *b\_num*: string)  
Profile (*SSN:* integer, p\_*name*: string, *b\_group*: string, *med\_report*: string, *addr*: string, *num*:

string)

Donates(*did:* integer, *b\_name:* string)

Collects(*pid:* integer, *b\_name:*string)

**1.1.3.2 Current Queries and Functional Dependencies:**

Patient (*pid*) P → S, in BCNF Form

(SP)

Donor (*did*) D → S, in BCNF Form

(SD)

Blood Blank (*b\_name*, *b\_address*, *b\_num*) N → AX  
 (NAX)

N → AX is not in BCNF

N → AX is in 3NF because there are no transitive dependencies in this case

Profile(*SSN*, p\_*name*, *b\_group*, *med\_report*, *addr*, *num*) S → NBREY

(SNBREY)

S → NBREY is not in 3NF because it has transitive dependencies (P → S and D→ S and

S → NBREY)

Therefore, S → NBREY is in 2NF

In order to put this back into 3NF, transitive dependencies must be removed and be

functionally dependent on the key → R1 = {P, D, S} and R2 = {S, N, B, R, E Y}.

Donates(*did*, *b\_name*): No FDs, in BCNF Form

Collects(*pid*, *b\_name*): No FDs, in BCNF Form

**1.2 DATABASE IMPLEMENTATION AND GUI DESIGN**

**1.2.1 Creating Tables and Inputting Data**

**CREATE TABLE Patients(**

**pid int PRIMARY KEY,**

**);**

**INSERT INTO Patients VALUES**

**(100),**

**(101),**

**(102),**

**(103),**

**(104),**

**(105),**

**(106),**

**(107),**

**(108),**

**(109),**

**(110),**

**(111),**

**(112),**

**(113)**

**/\*table for Donor\*/**

**CREATE TABLE Donors(**

**did int PRIMARY KEY,**

**);**

**INSERT INTO Donors VALUES**

**(200),**

**(201),**

**(202),**

**(203),**

**(204),**

**(205),**

**(206),**

**(207),**

**(208),**

**(209)**

**/\*table for Profiles\*/**

**CREATE TABLE Profiles(**

**SSN int PRIMARY KEY,**

**p\_name varchar(50),**

**b\_group varchar(50),**

**med\_report varchar(50),**

**addr varchar(50),**

**num varchar(10),**

**);**

**INSERT INTO Profiles VALUES**

**(312085335, 'Karen Mani', 'AB+', 'hemophilia', '123 Cherry Lane', '2027857347'),**

**(549330490, 'Yevgeny Smolyansky', 'O-', NULL, '263 Berwind Road', '6103899547'),**

**(473849203, 'Ethan Smith', 'A+', NULL, '1600 Penn Ave', '2027383743'),**

**(423912947, 'John Doe', 'O-', 'anemia', '2100 F St NW', '3567283647'),**

**(782584039, 'Sarah Smith', 'B-', 'diabetes', '1900 F St NW', '4672938989'),**

**(187048593, 'Jack Black', 'O-', NULL, '42 Wallaby Way', '2025647122'),**

**(672957384, 'Lyndsay Goldstein', 'A-', 'anxiety', '787 I St', '7839238484'),**

**(957204930, 'Mary Jane', 'AB+', 'brain bleed', '78 Oak Lane', '9004392323'),**

**(111111111, 'Hal Irish', 'B+', NULL, '672 Peach Tree Road', '8006102022'),**

**(121212121, 'Jeremiah Smith', 'O+', 'hypochondria', '808 Sesame St', '9093237483'),**

**(131658484, 'Warby Parker', 'A-', 'blind', '56 Beech Tree Road', '6009871234'),**

**(432824940, 'Stephen Morris', 'AB-', NULL, '15 Pine St', '7817295431'),**

**(284582403, 'Hannah Glaser', 'B-', 'POTS', '2400 M St', '3108905678'),**

**(185939040, 'Kennedy Young', 'AB+', 'pregnant', '78 A St', '9087789672'),**

**(120583039, 'Lily Jacobs', 'O-', NULL, '678 Kensico St', '9084328743'),**

**(420548205, 'John Hudson', 'AB+', 'HIV', '10 Ocean Drive', '3657646732'),**

**(104492049, 'Joseph Parks', 'O+', NULL, '4 23rd St', '9098473241'),**

**(194803492, 'Julia Roberts', 'AB-', 'HPV', '23 Beverly Hills Dr', '5637281234'),**

**(238404025, 'Sasha Smolyansky', 'O+', NULL, '469 Merriweather Road', '3920483029'),**

**(830248022, 'Lei He', 'B-', NULL, '800 21 St NW', '5739294574'),**

**(955832433, 'Jesse McCartney', 'A+', 'AIDS', '123 Hive Road', '9894620876'),**

**(197639573, 'Estelle Gelman', 'AB-', 'high blood pressure', '2100 H St NW', '2029945609')**

**CREATE TABLE BloodBanks(**

**b\_name varchar(100) PRIMARY KEY,**

**b\_address varchar(50),**

**b\_num varchar(50),**

**);**

**INSERT INTO BloodBanks VALUES**

**('American Red Cross Blood Donation Center', '1900 Penn Ave', '2026758940'),**

**('Hong Kong Red Cross Blood Transfusion Service', '200 3rd St', '8902389393'),**

**('Inova Blood Donor Services', '23 F St', '9098765432'),**

**('Blood Donation Center', '1 Central Park Ave', '6009006767')**

**CREATE TABLE Donate(**

**pid int,**

**b\_name varchar(100),**

**FOREIGN KEY(pid) REFERENCES Patients,**

**FOREIGN KEY(b\_name) REFERENCES BloodBanks,**

**PRIMARY KEY(pid, b\_name),**

**);**

**INSERT INTO Donate VALUES**

**(101, 'Blood Donation Center'),**

**(102, 'Blood Donation Center'),**

**(103, 'Blood Donation Center'),**

**(107, 'Hong Kong Red Cross Blood Transfusion Service'),**

**(108, 'Hong Kong Red Cross Blood Transfusion Service'),**

**(109, 'American Red Cross Blood Donation Center')**

**CREATE TABLE Collect(**

**did int,**

**b\_name varchar(100),**

**FOREIGN KEY(did) REFERENCES Donors,**

**FOREIGN KEY(b\_name) REFERENCES BloodBanks,**

**PRIMARY KEY(did, b\_name),**

**);**

**INSERT INTO Collect VALUES**

**(200, 'Blood Donation Center'),**

**(201, 'Blood Donation Center'),**

**(202, 'Blood Donation Center'),**

**(203, 'American Red Cross Blood Donation Center'),**

**(204, 'Blood Donation Center'),**

**(207, 'Blood Donation Center')**

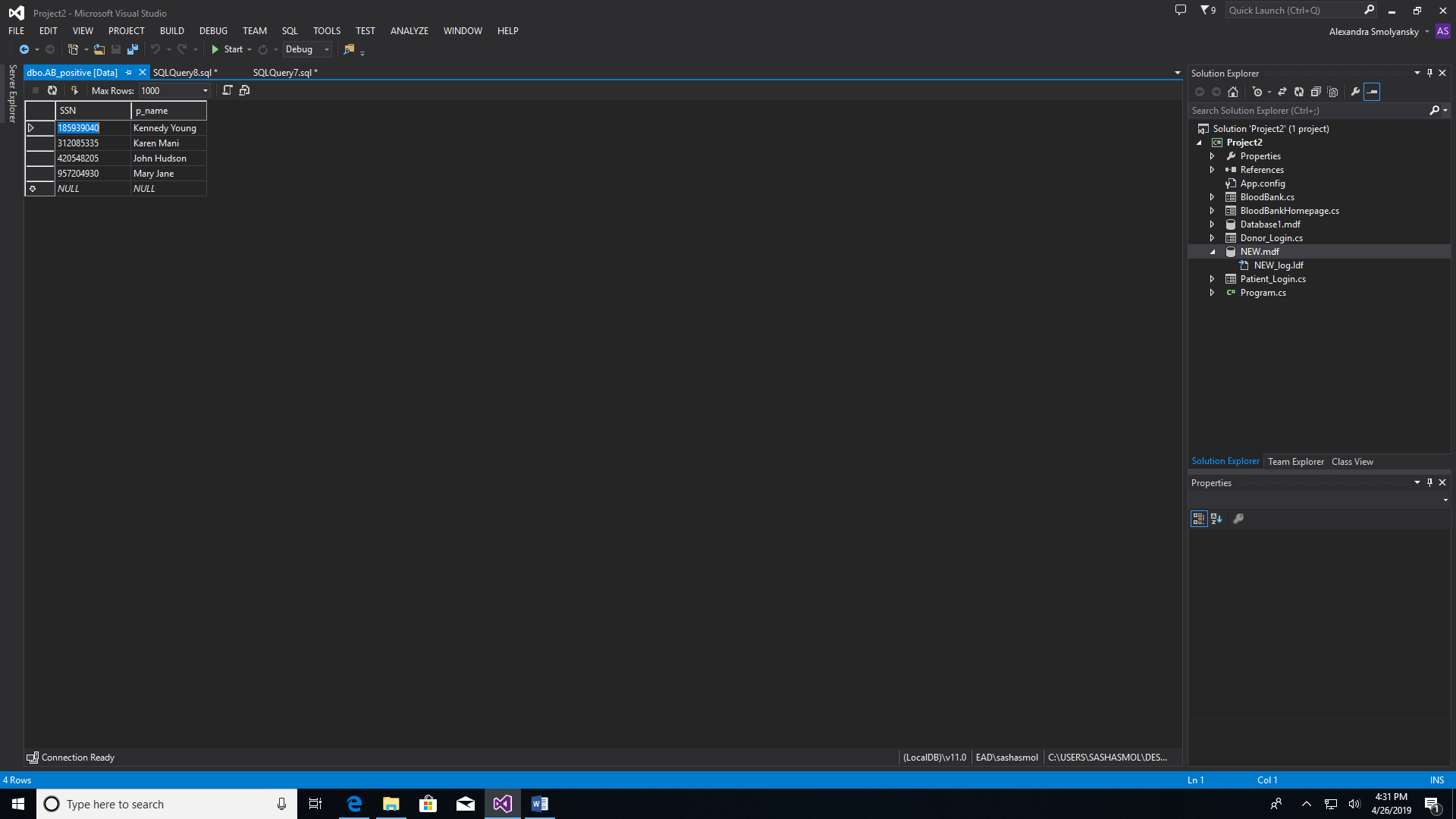
**1.2.2 Views NEW**

**CREATE VIEW AB\_positive**

**AS SELECT SSN,p\_name**

**FROM Profiles**

**WHERE b\_group = 'AB+';**

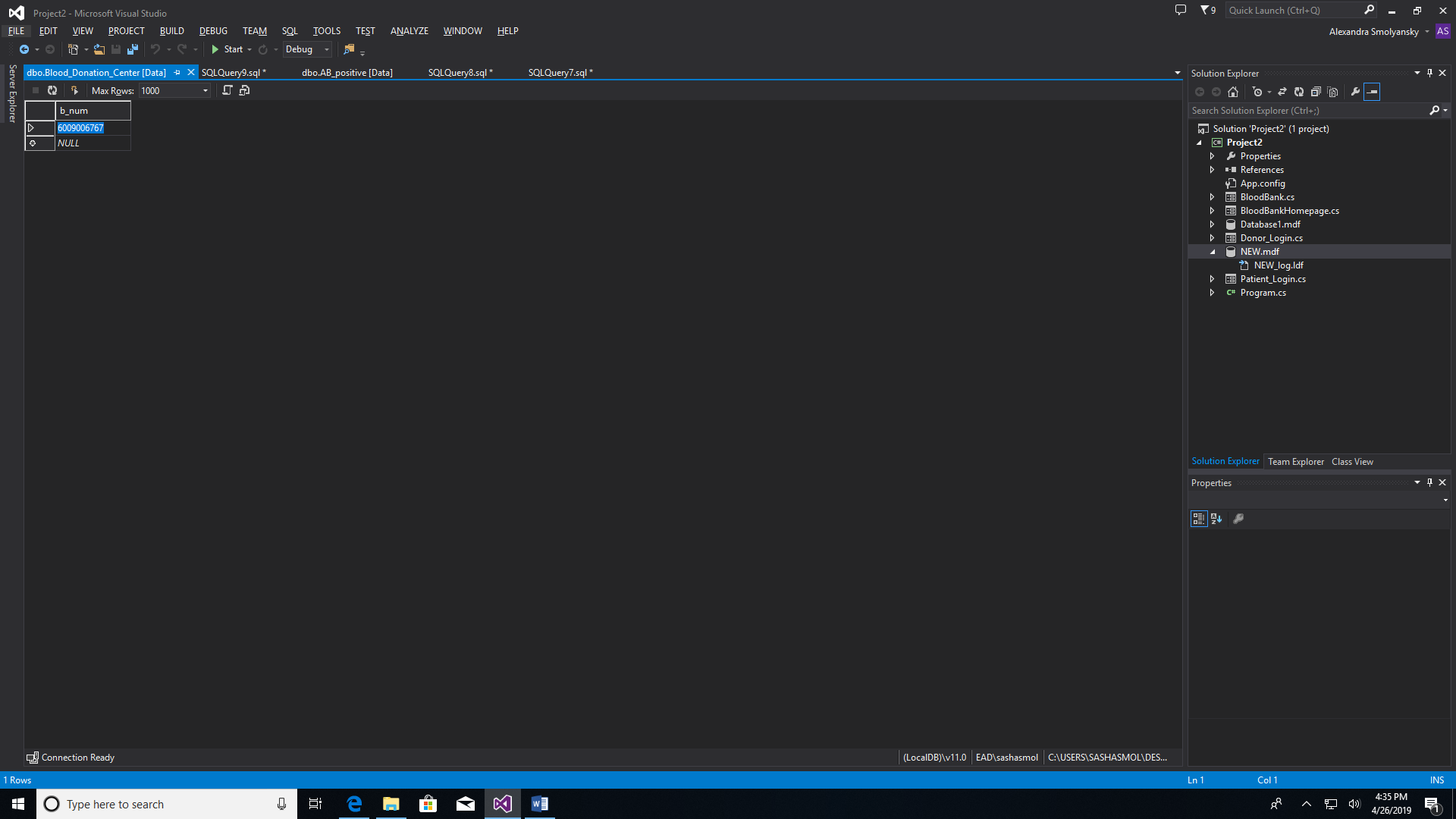
****

**CREATE VIEW Blood\_Donation\_Center**

**AS SELECT b\_num**

**FROM BloodBanks**

**WHERE b\_name = 'Blood Donation Center';**

****

**1.2.3 Triggers NEW**

**CREATE TABLE op\_log (op\_type varchar(10), op\_time datetime);**

**CREATE TRIGGER op\_update**

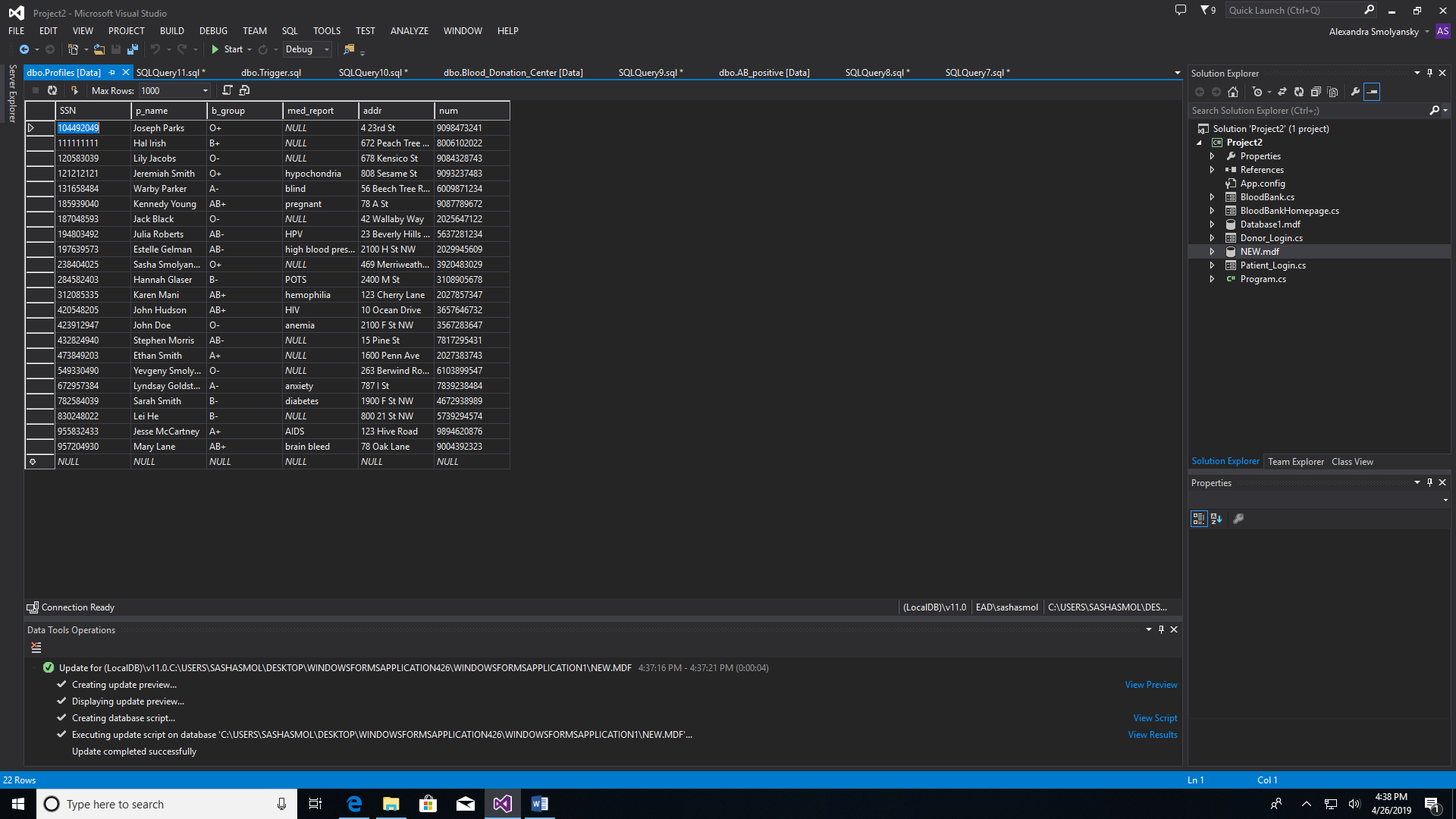
**ON BloodBanks**

**FOR UPDATE**

**AS**

**INSERT INTO op\_log VALUES('Update', GETDATE());**

**UPDATE Profiles SET p\_name = 'Mary Lane' where p\_name = 'Mary Jane';**

**<-- last line**

**CREATE TRIGGER op\_delete**

**ON Donors**

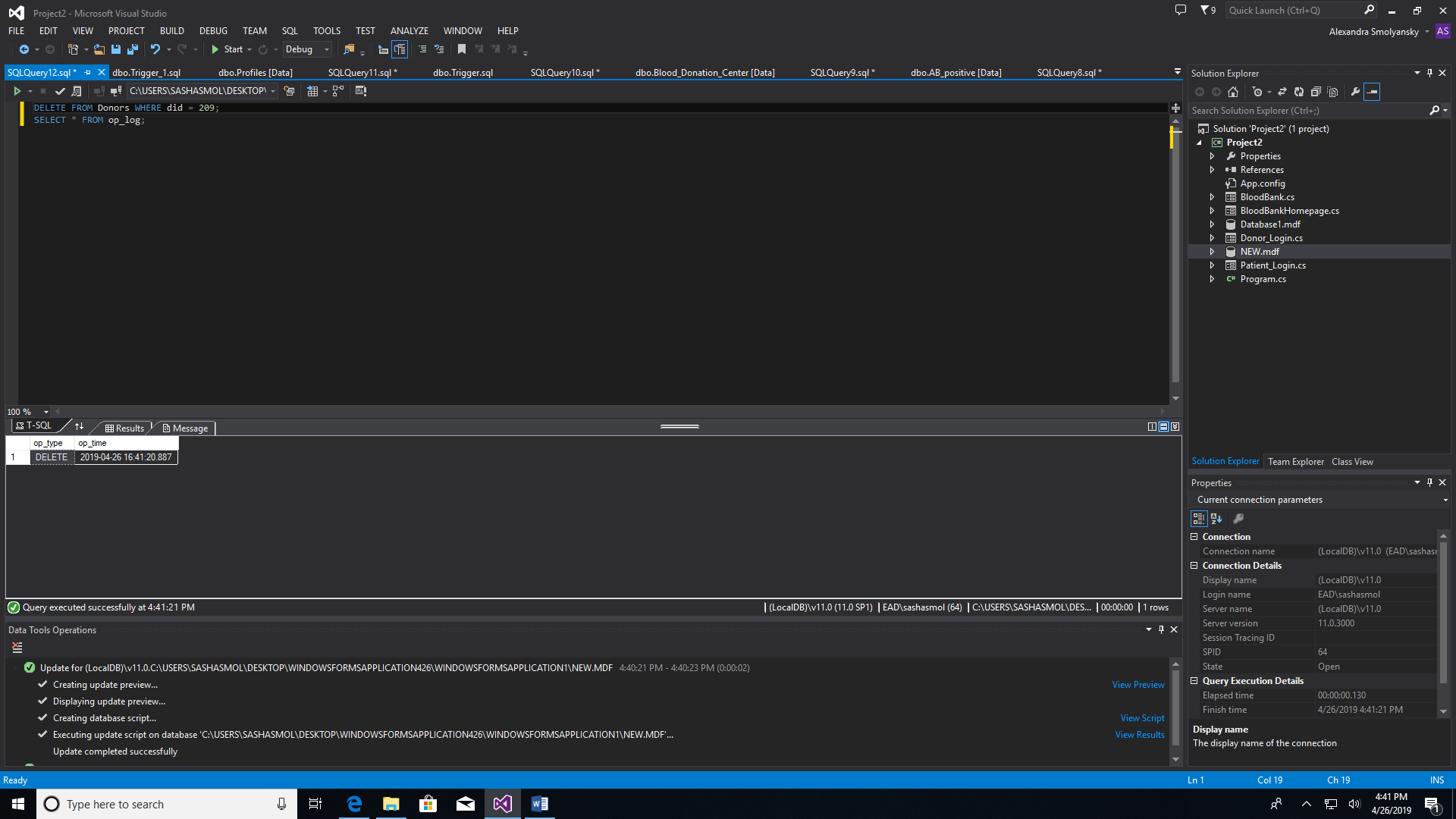
**FOR DELETE**

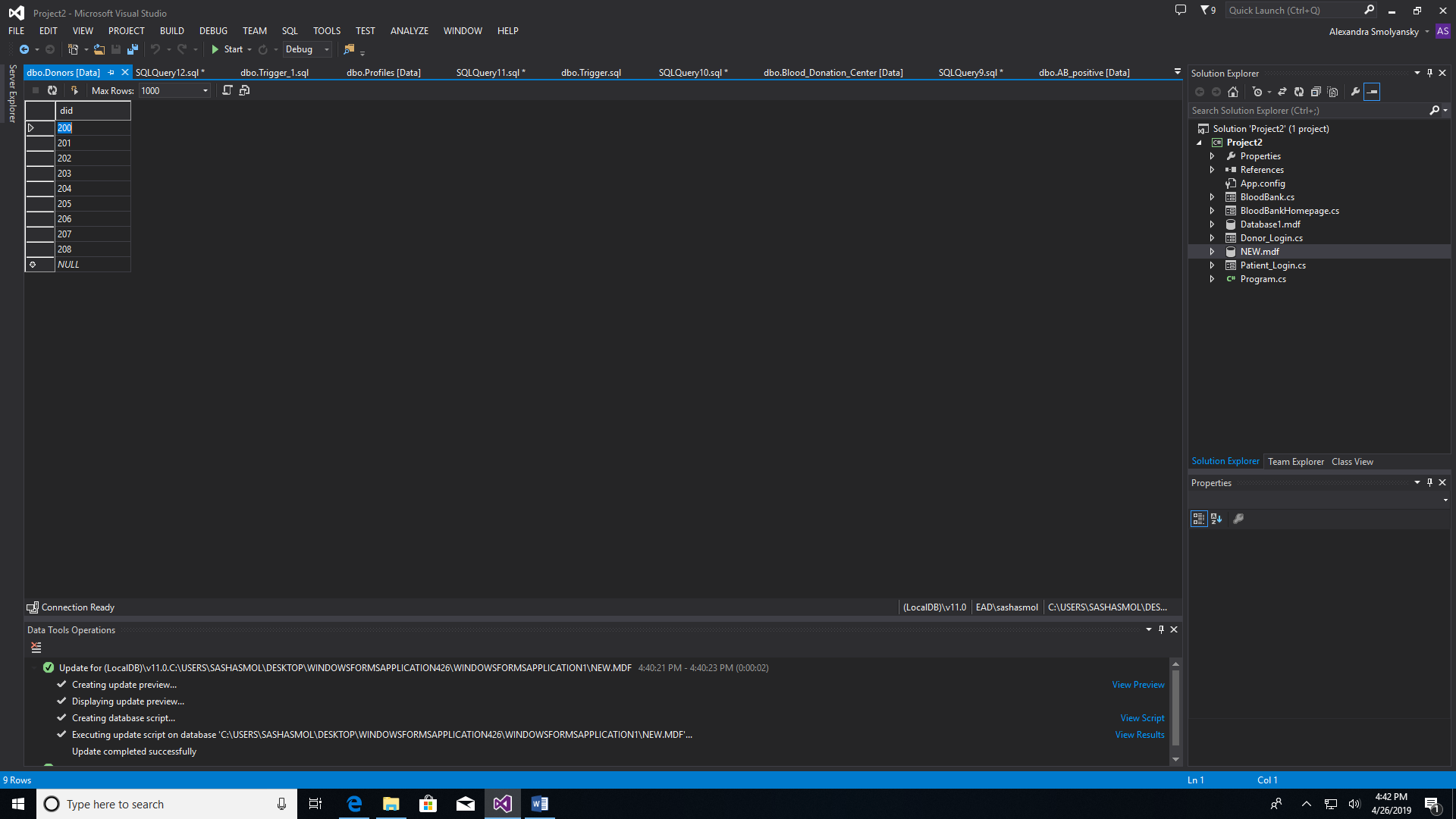
**AS**

**INSERT INTO op\_log VALUES('DELETE', GETDATE());**

**DELETE FROM Donors WHERE did = 209;**

**SELECT \* FROM op\_log;**

****

**<-- 209 is gone**

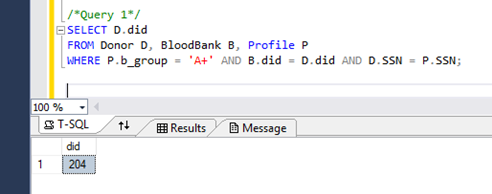
**1.2.4 Twenty Queries from Project 1**

The queries were done with the design of our original database which we ended up updating. Please keep in mind that the data/methods used to project the specific information has been modified which is why it might not necessarily match.

1. SELECT D.did

FROM Donor D, BloodBank B, Profile P

WHERE P.b\_group = 'A+' AND B.did = D.did AND D.SSN = P.SSN;



1. SELECT D.did

FROM Donor D, BloodBank B, Profile P

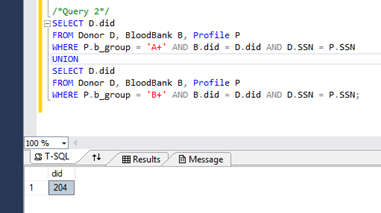
WHERE P.b\_group = 'A+' AND B.did = D.did AND D.SSN = P.SSN

UNION

SELECT D.did

FROM Donor D, BloodBank B, Profile P

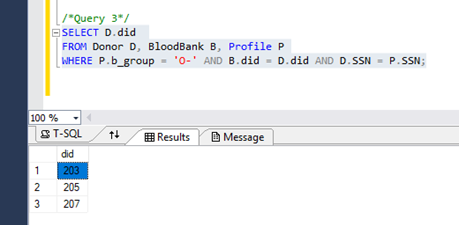
WHERE P.b\_group = 'B+' AND B.did = D.did AND D.SSN = P.SSN;



1. SELECT D.did

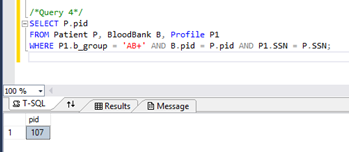
FROM Donor D, BloodBank B, Profile P

WHERE P.b\_group = 'O-' AND B.did = D.did AND D.SSN = P.SSN;



1. SELECT P.pid

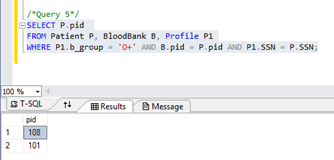
FROM Patient P, BloodBank B, Profile P1

WHERE P1.b\_group = 'AB+' AND B.pid = P.pid AND P1.SSN = P.SSN;

1. SELECT P.pid

FROM Patient P, BloodBank B, Profile P1

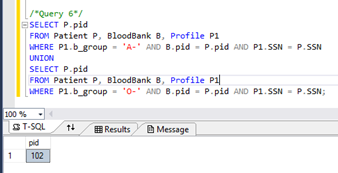
WHERE P1.b\_group = 'O+' AND B.pid = P.pid AND P1.SSN = P.SSN;



1. SELECT P.pid

FROM Patient P, BloodBank B, Profile P1

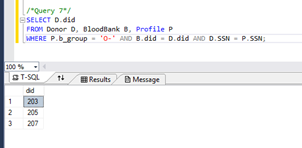
WHERE P1.b\_group = 'A-' AND B.pid = P.pid AND P1.SSN = P.SSN



1. SELECT D.did

FROM Donor D, BloodBank B, Profile P

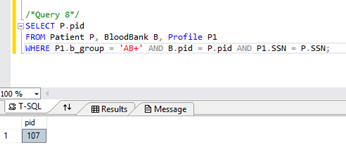
WHERE P.b\_group = 'O-' AND B.did = D.did AND D.SSN = P.SSN;



1. SELECT P.pid

FROM Patient P, BloodBank B, Profile P1

WHERE P1.b\_group = 'AB+' AND B.pid = P.pid AND P1.SSN = P.SSN;



1. SELECT P.pid, P1.p\_name

FROM Patient P, BloodBank B, Profile P1

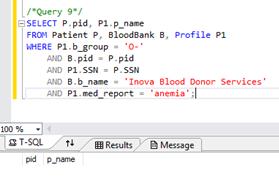
WHERE P1.b\_group = 'O-'

AND B.pid = P.pid

AND P1.SSN = P.SSN

AND B.b\_name = 'Inova Blood Donor Services'

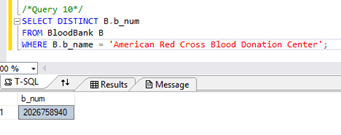
AND P1.med\_report = 'anemia';



1. SELECT DISTINCT B.b\_num

FROM BloodBank B

WHERE B.b\_name = 'American Red Cross Blood Donation Center';



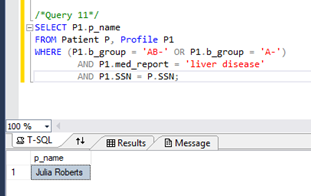
1. SELECT P1.p\_name

FROM Patient P, Profile P1

WHERE (P1.b\_group = 'AB-' OR P1.b\_group = 'A-')

AND P1.med\_report = 'liver disease'

AND P1.SSN = P.SSN;



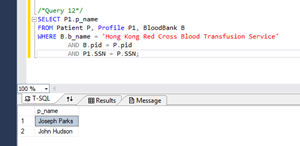
1. SELECT P1.p\_name

FROM Patient P, Profile P1, BloodBank B

WHERE B.b\_name = 'Hong Kong Red Cross Blood Transfusion Service'

AND B.pid = P.pid

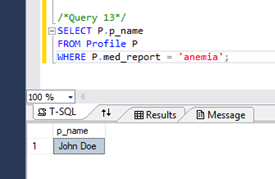
AND P1.SSN = P.SSN;



1. SELECT P.p\_name

FROM Profile P

WHERE P.med\_report = 'anemia';



1. SELECT B.b\_address

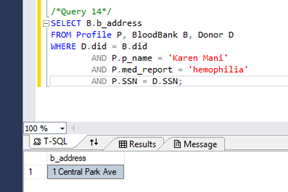
FROM Profile P, BloodBank B, Donor D

WHERE D.did = B.did

AND P.p\_name = 'Karen Mani'

AND P.med\_report = 'hemophilia'

AND P.SSN = D.SSN;



1. SELECT X.pid

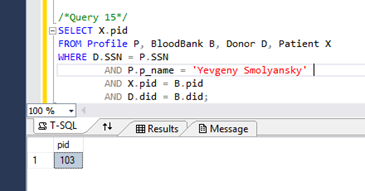
FROM Profile P, BloodBank B, Donor D, Patient X

WHERE D.SSN = P.SSN

AND P.p\_name = 'Yevgeny Smolyansky'

AND X.pid = B.pid

AND D.did = B.did;

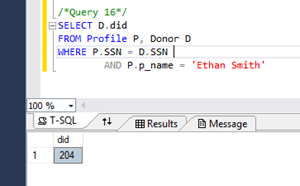


1. SELECT D.did

FROM Profile P, Donor D

WHERE P.SSN = D.SSN

AND P.p\_name = 'Ethan Smith'



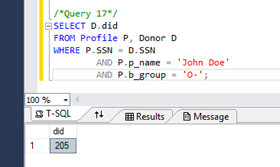
1. SELECT D.did

FROM Profile P, Donor D

WHERE P.SSN = D.SSN

AND P.p\_name = 'John Doe'

AND P.b\_group = 'O-';



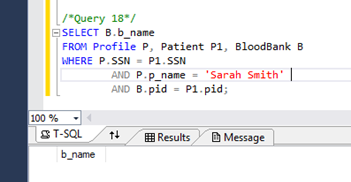
1. SELECT B.b\_name

FROM Profile P, Patient P1, BloodBank B

WHERE P.SSN = P1.SSN

AND P.p\_name = 'Sarah Smith'

AND B.pid = P1.pid;



1. SELECT B.b\_name

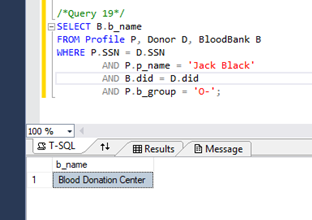
FROM Profile P, Donor D, BloodBank B

WHERE P.SSN = D.SSN

AND P.p\_name = 'Jack Black'

AND B.did = D.did

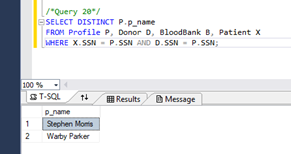
AND P.b\_group = 'O-';



1. SELECT DISTINCT P.p\_name

FROM Profile P, Donor D, BloodBank B, Patient X

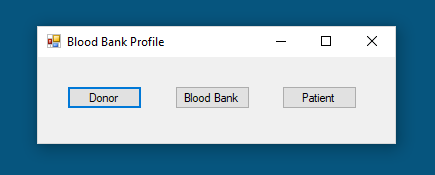
WHERE X.SSN = P.SSN AND D.SSN = P.SSN;

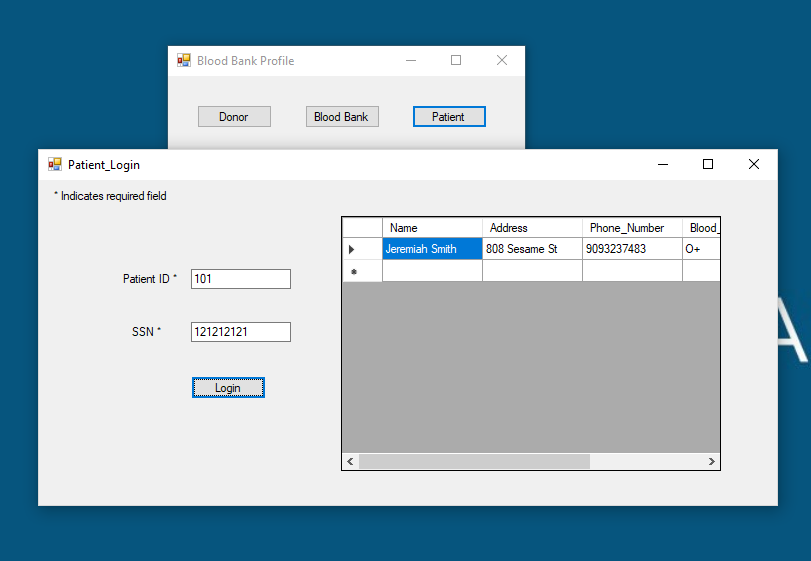
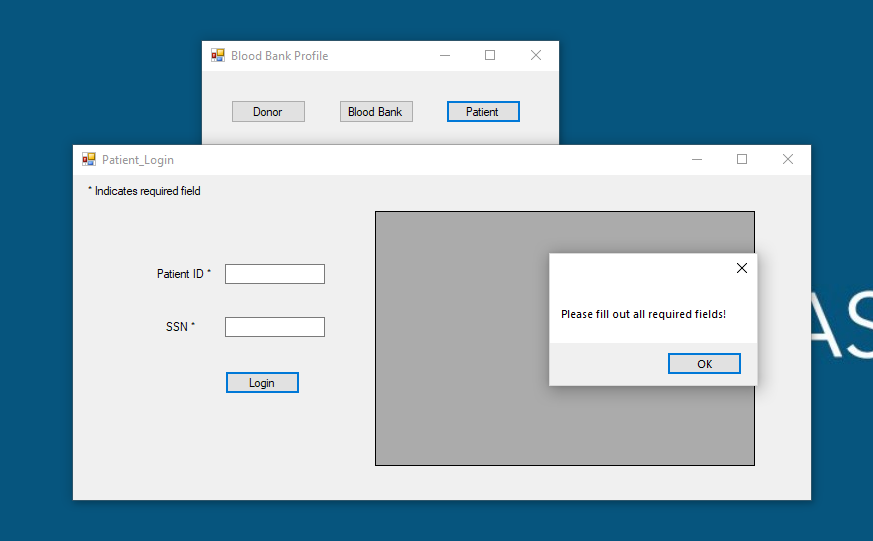


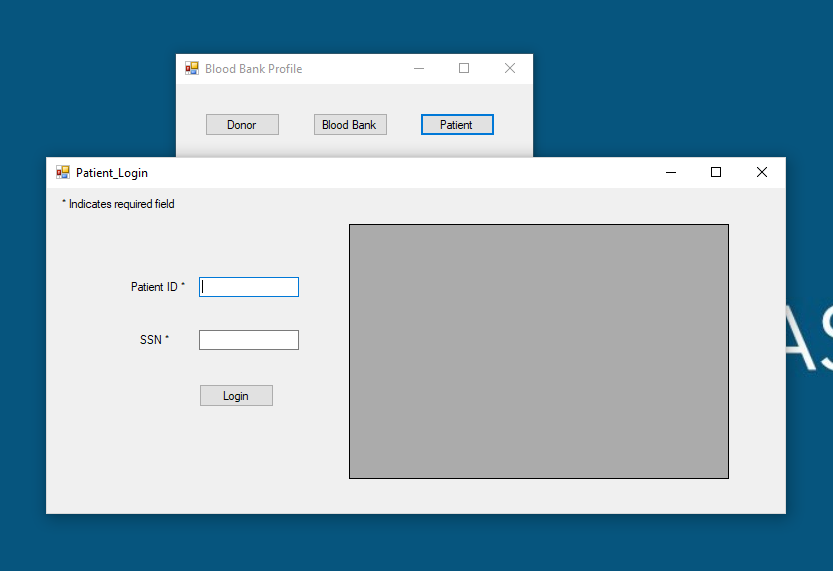
**1.2.5 GUI Design**

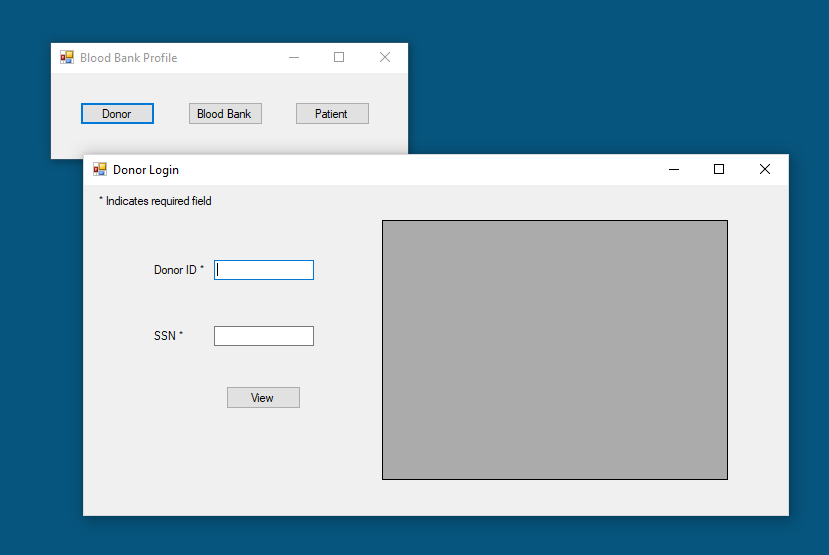
**1.2.5.1 Images of the Database**

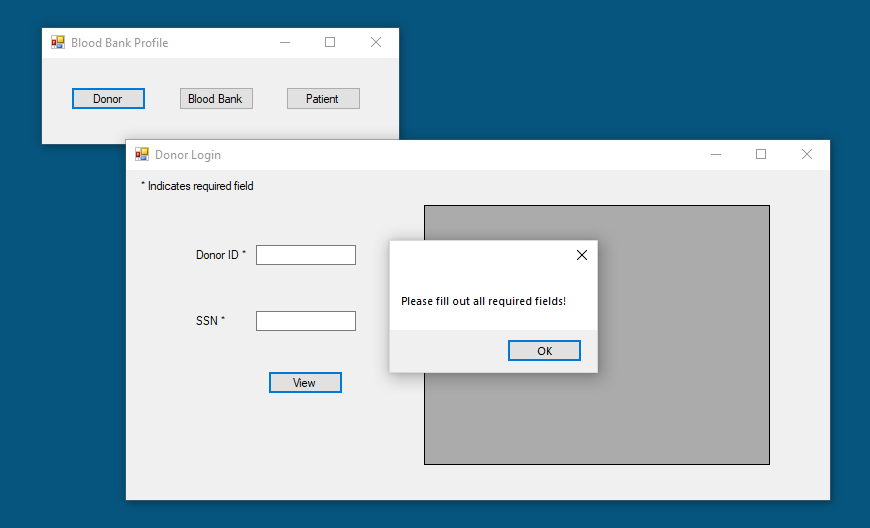
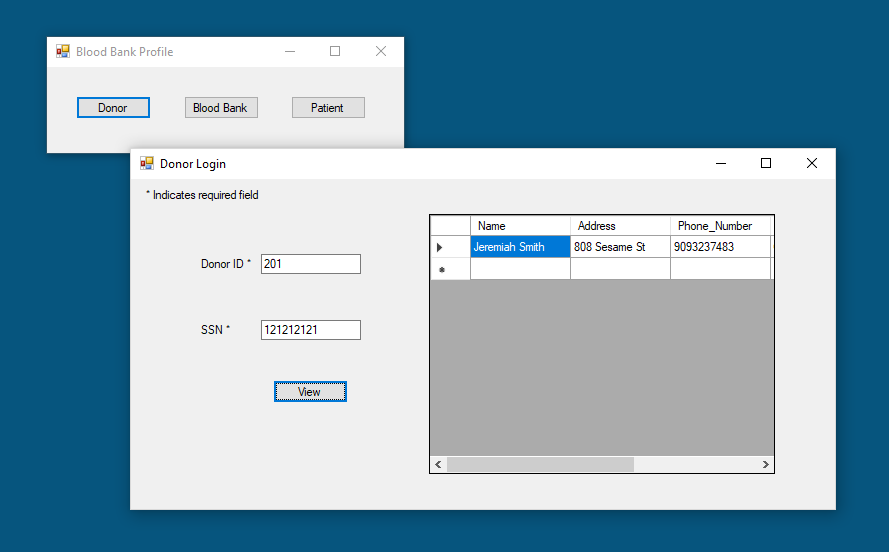
*Homepage*



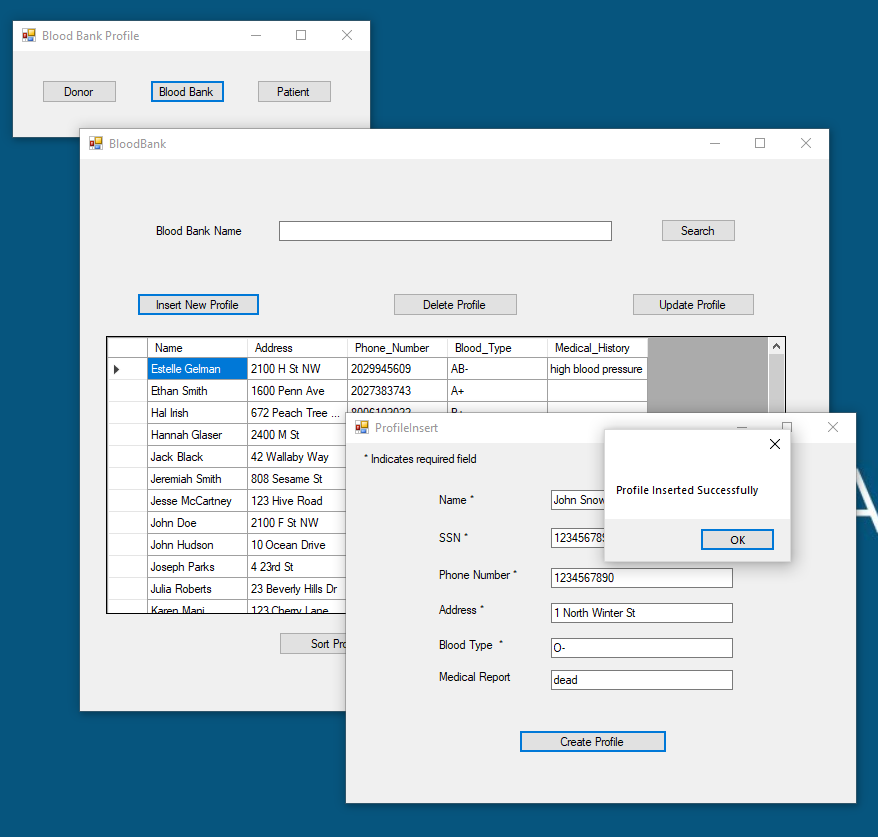
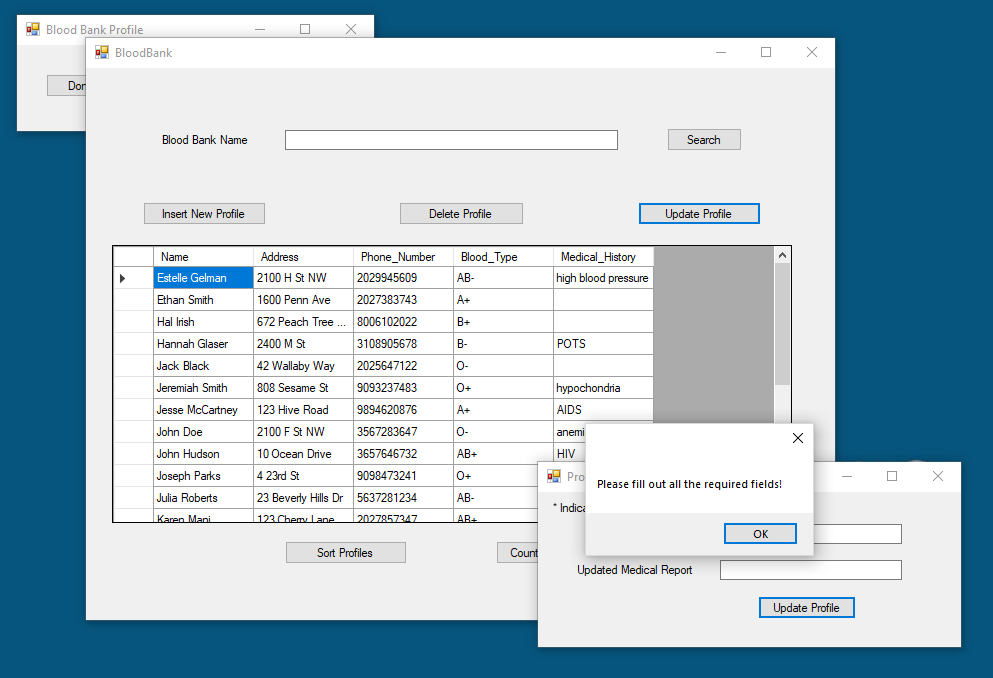
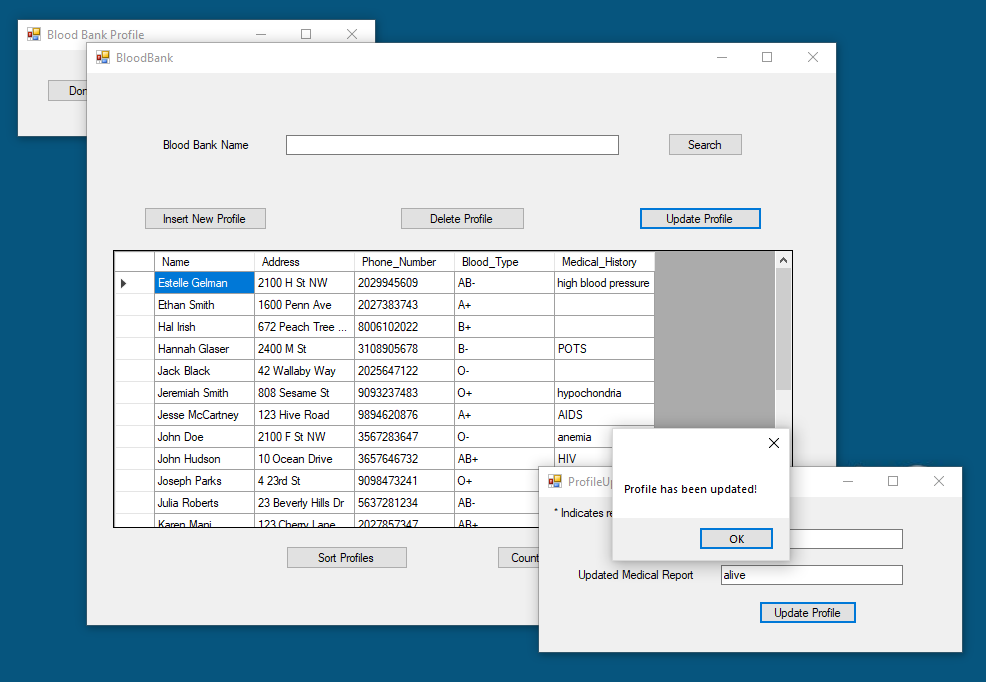
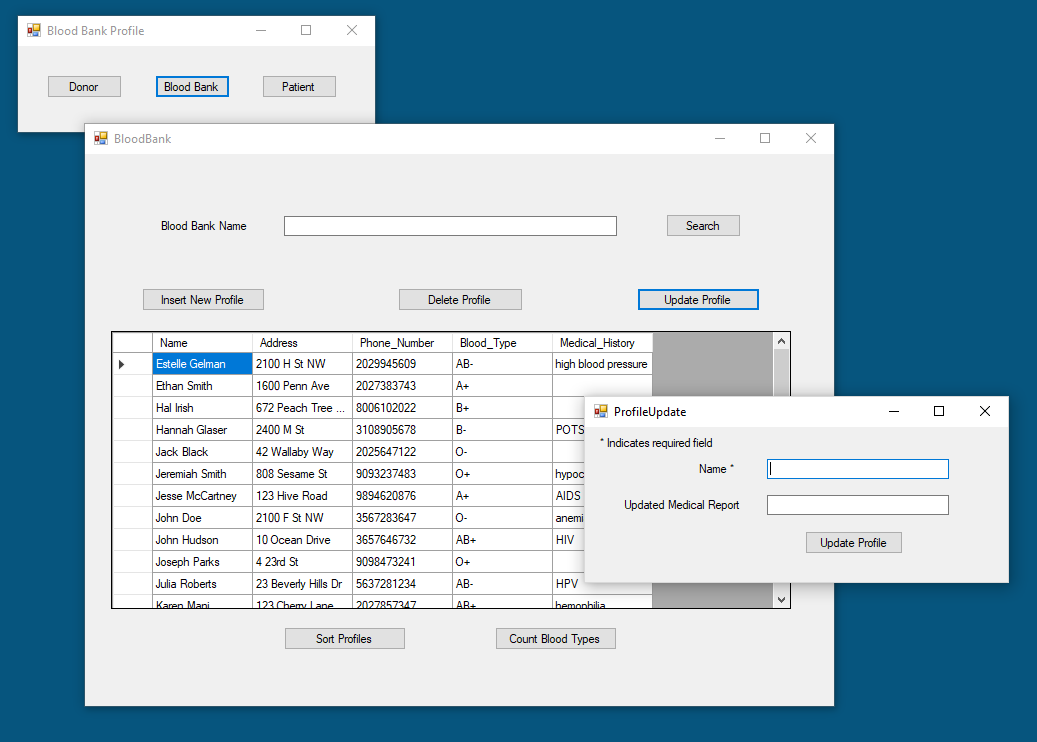
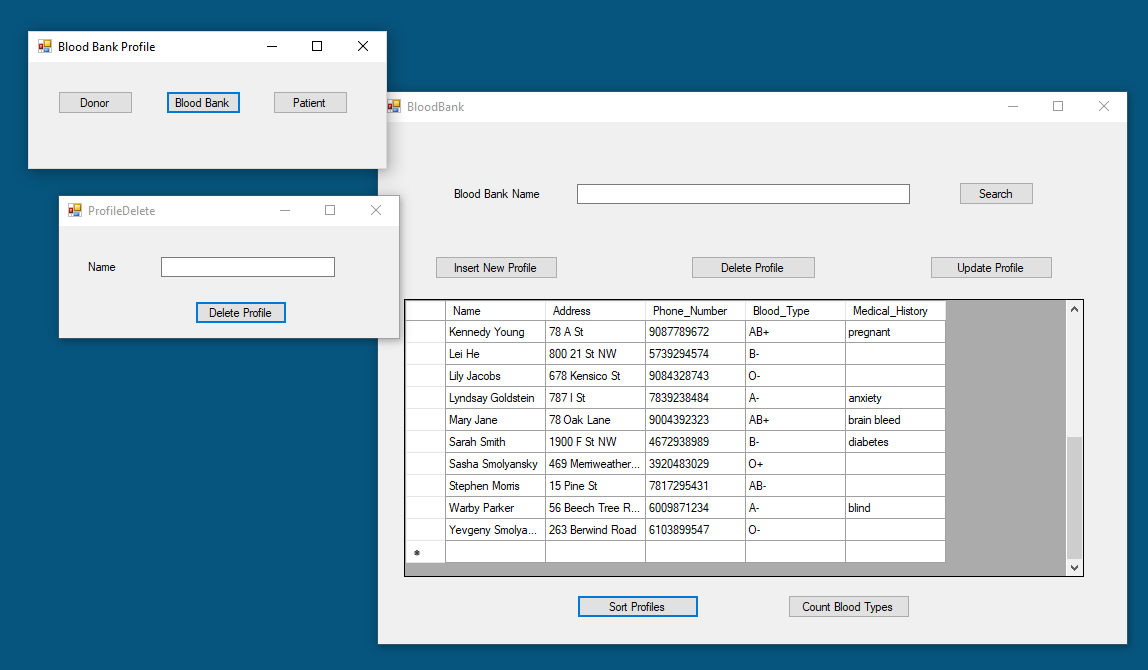
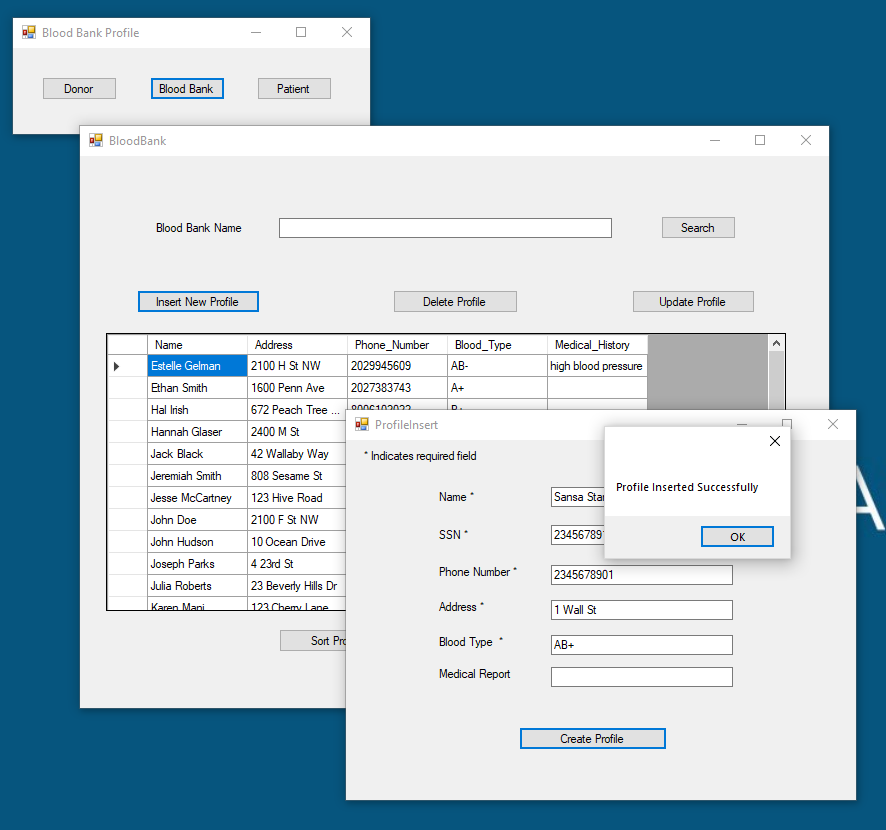
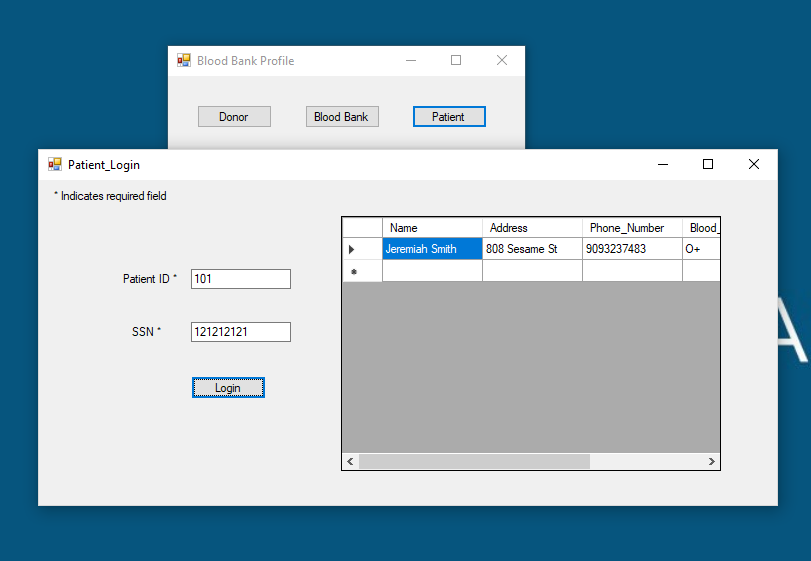
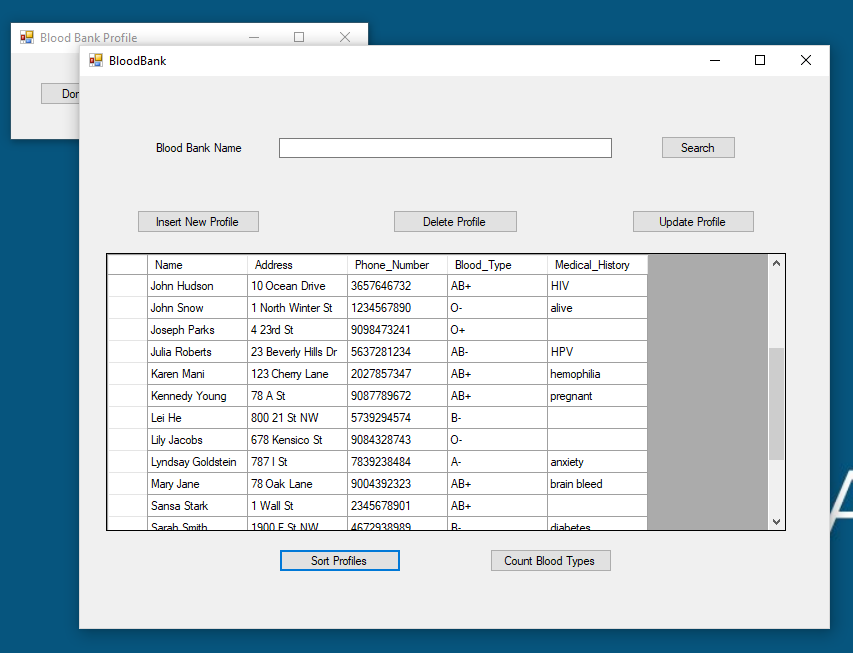
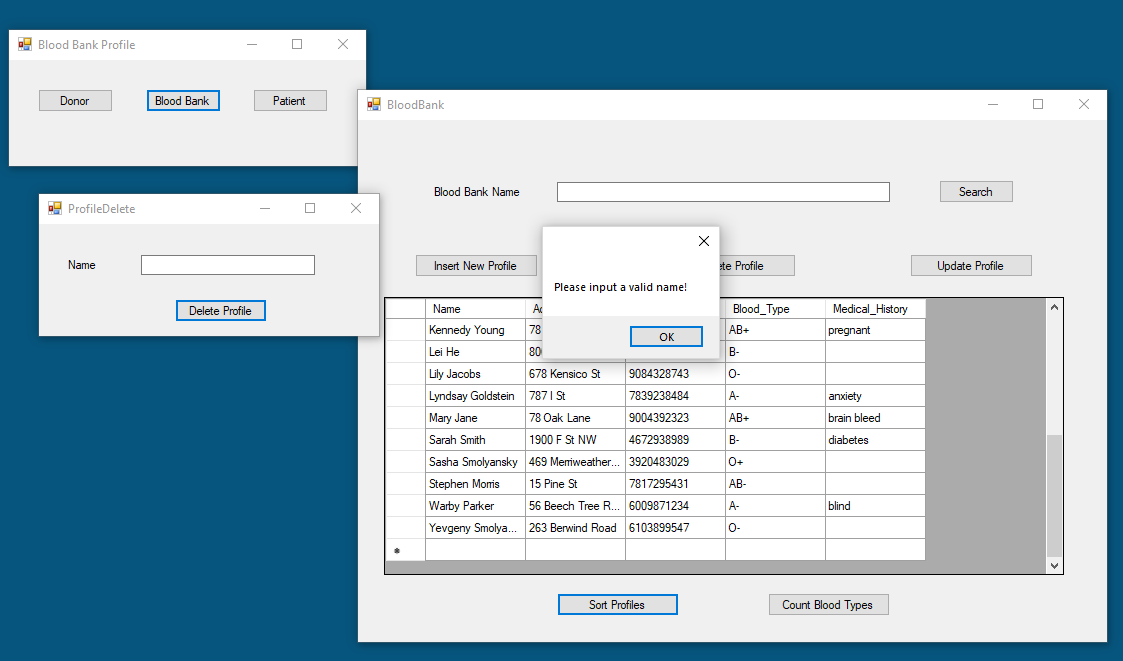
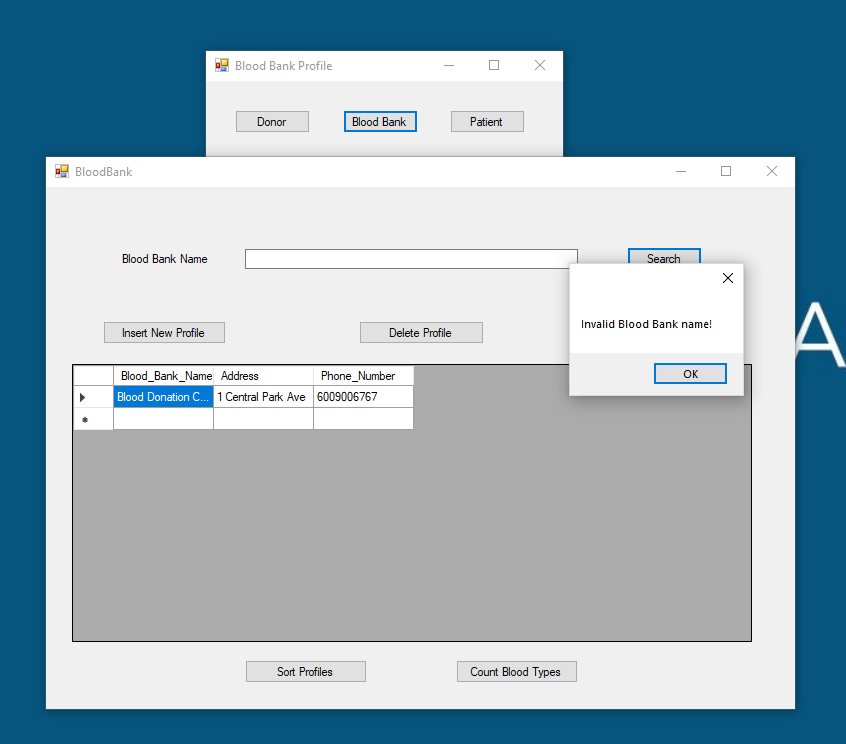
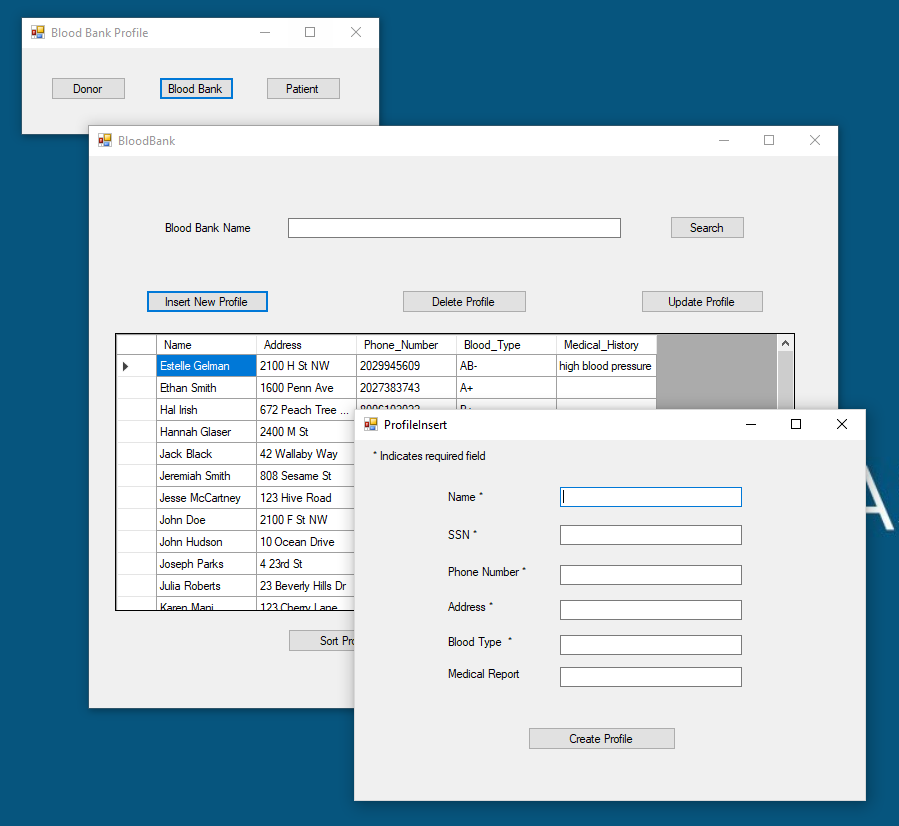
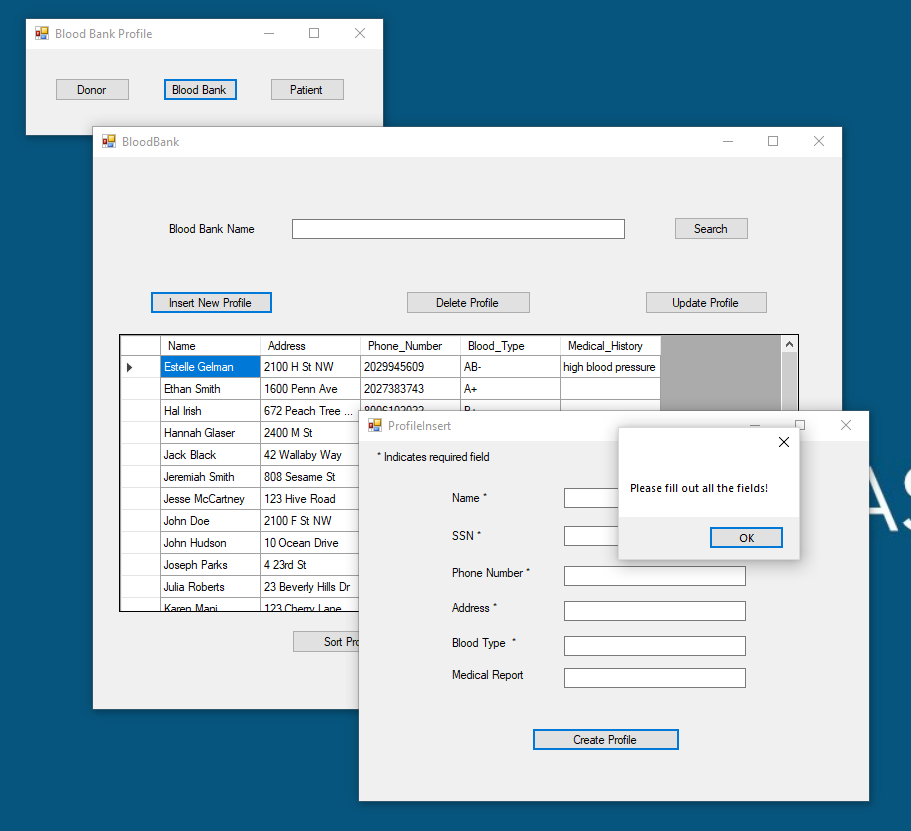
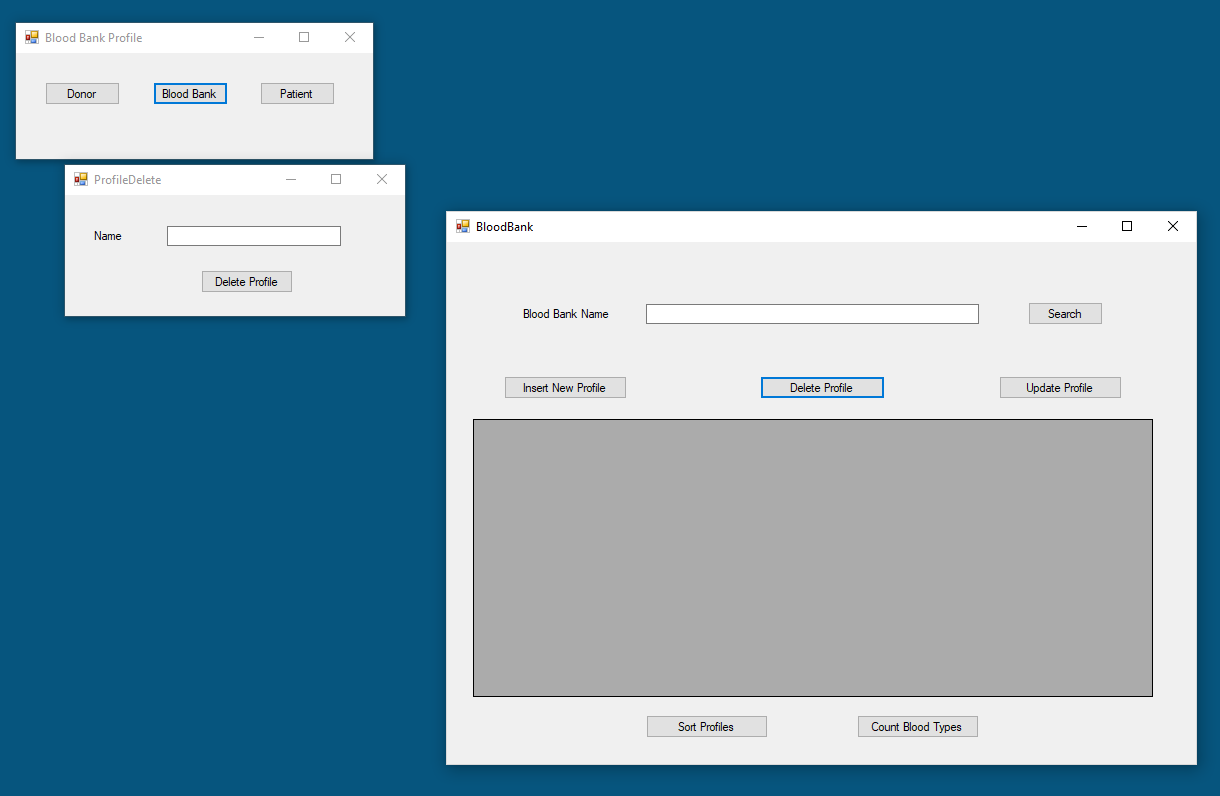
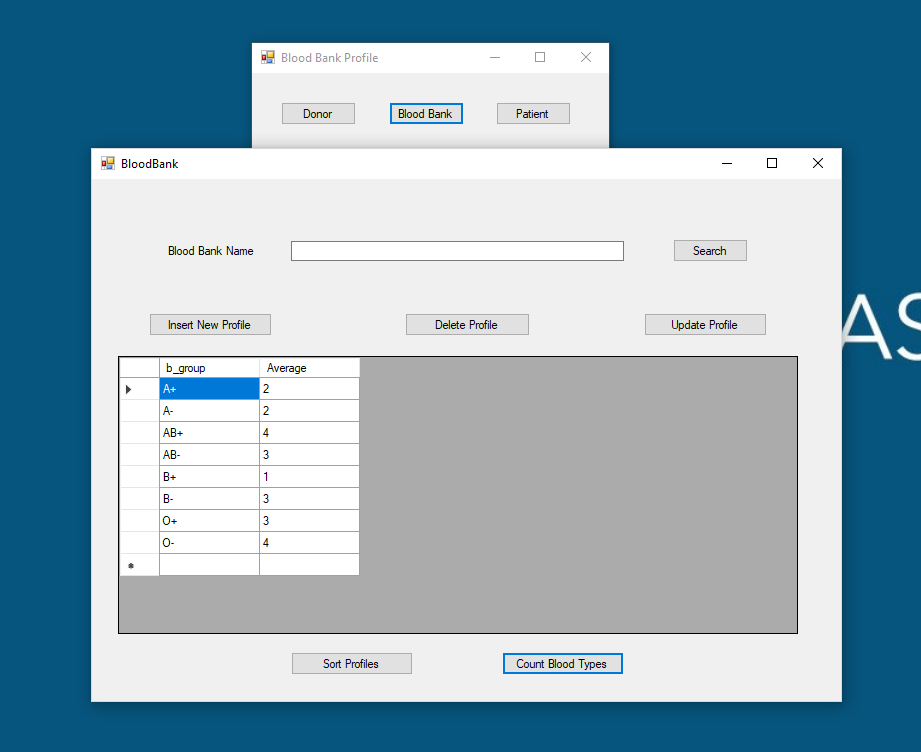
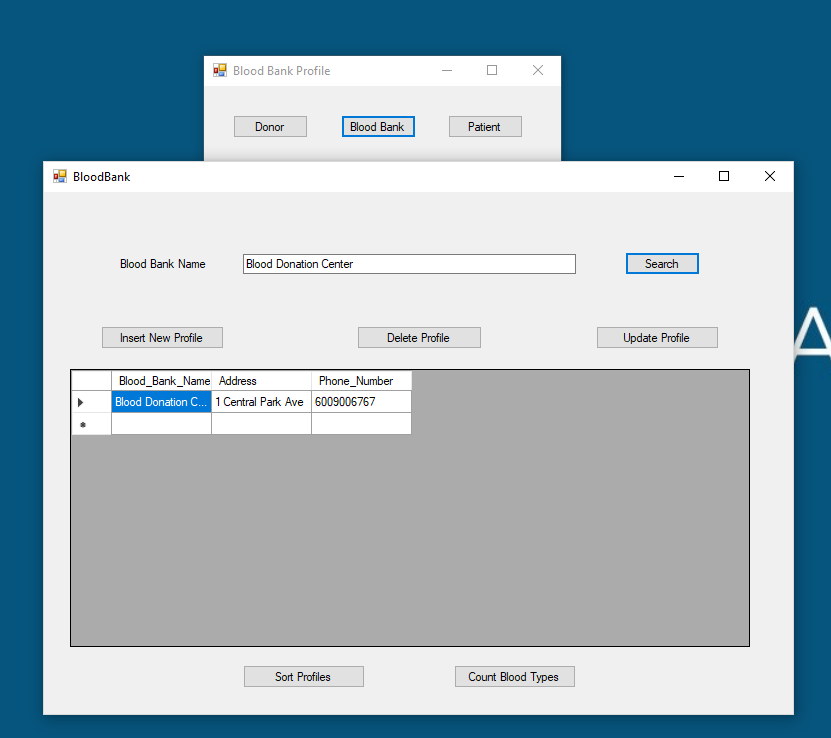
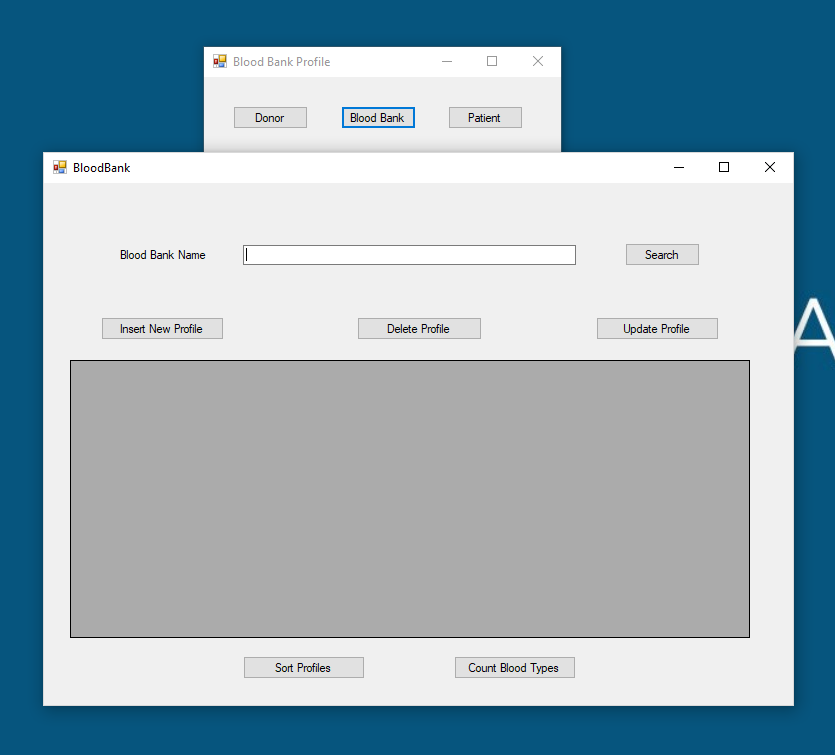
*Patient Login*

**

*Donor Login*

**

*Blood Bank*

**

**1.2.5.2 Database Code**

**Blood Bank Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**using System.Data.SqlClient;**

**namespace WindowsFormsApplication1**

**{**

**public partial class BloodBank : Form**

**{**

**SqlConnection con3 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlConnection con4 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd3;**

**SqlCommand cmd4;**

**SqlConnection con5 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd5;**

**SqlConnection con6 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd6;**

**SqlConnection con7 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd7;**

**public BloodBank()**

**{**

**InitializeComponent();**

**}**

**private void label1\_Click(object sender, EventArgs e)**

**{**

**}**

**private void show\_data()**

**{**

**con3.Open();**

**SqlCommand cmd3 = new SqlCommand("SELECT DISTINCT B.b\_name AS Blood\_Bank\_Name, B.b\_num AS Phone\_Number, B.b\_address AS Address FROM BloodBanks B WHERE B.b\_name = @b\_name", con3);**

**cmd3.Parameters.AddWithValue("@b\_name", textBox\_bloodbankname.Text);**

**SqlDataAdapter adapt5 = new SqlDataAdapter(cmd3);**

**DataTable dt5 = new DataTable();**

**adapt5.SelectCommand = cmd3;**

**adapt5.Fill(dt5);**

**dataGridView2.DataSource = dt5;**

**con3.Close();**

**}**

**private void button\_view\_Click(object sender, EventArgs e)**

**{**

**if (textBox\_bloodbankname.Text == "")**

**{**

**MessageBox.Show("Invalid Blood Bank name!");**

**return;**

**}**

**show\_data();**

**}**

**private void button\_sort\_Click(object sender, EventArgs e)**

**{**

**con4.Open();**

**SqlCommand cmd4 = new SqlCommand("SELECT DISTINCT P.p\_name AS Name, P.addr AS Address, P.num AS Phone\_Number, P.b\_group AS Blood\_Type, P.med\_report AS Medical\_History FROM Profiles P, Patients X, Donors Y, Collect C, Donate D WHERE D.pid = X.pid AND C.did = Y.did ORDER BY P.p\_name ASC", con4);**

**SqlDataAdapter adapt1 = new SqlDataAdapter(cmd4);**

**DataTable dt1 = new DataTable();**

**adapt1.SelectCommand = cmd4;**

**adapt1.Fill(dt1);**

**dataGridView2.DataSource = dt1;**

**con4.Close();**

**}**

**private void button\_average\_Click(object sender, EventArgs e)**

**{**

**con6.Open();**

**SqlCommand cmd6 = new SqlCommand("SELECT P.b\_group, COUNT(\*) AS Average FROM Profiles P GROUP BY P.b\_group", con6);**

**SqlDataAdapter ad = new SqlDataAdapter(cmd6);**

**DataTable d = new DataTable();**

**ad.SelectCommand = cmd6;**

**ad.Fill(d);**

**dataGridView2.DataSource = d;**

**con6.Close();**

**}**

**private void textBox\_bloodbankname\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**private void button\_insert\_Click(object sender, EventArgs e)**

**{**

**ProfileInsert prof = new ProfileInsert();**

**prof.Show();**

**}**

**private void button\_delete\_Click(object sender, EventArgs e)**

**{**

**ProfileDelete del = new ProfileDelete();**

**del.Show();**

**}**

**private void button\_update\_Click(object sender, EventArgs e)**

**{**

**ProfileUpdate upd = new ProfileUpdate();**

**upd.Show();**

**}**

**}**

**}**

**Blood Bank Homepage Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace WindowsFormsApplication1**

**{**

**public partial class BloodBankHomepage : Form**

**{**

**public BloodBankHomepage()**

**{**

**InitializeComponent();**

**}**

**private void label1\_Click(object sender, EventArgs e)**

**{**

**}**

**private void Form1\_Load(object sender, EventArgs e)**

**{**

**}**

**private void button\_login\_Click(object sender, EventArgs e)**

**{**

**Patient\_Login pnt = new Patient\_Login();**

**pnt.Show();**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**Donor\_Login dnr = new Donor\_Login();**

**dnr.Show();**

**}**

**private void button\_bloodbank\_Click(object sender, EventArgs e)**

**{**

**BloodBank bbank = new BloodBank();**

**bbank.Show();**

**}**

**}**

**}**

**Donor Login Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**using System.Data.SqlClient;**

**namespace WindowsFormsApplication1**

**{**

**public partial class Donor\_Login : Form**

**{**

**SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd;**

**public Donor\_Login()**

**{**

**InitializeComponent();**

**}**

**private void label1\_Click(object sender, EventArgs e)**

**{**

**}**

**private void show\_data()**

**{**

**con.Open();**

**SqlCommand cmd = new SqlCommand("SELECT P.p\_name AS Name, P.addr AS Address, P.num AS Phone\_Number, P.b\_group AS Blood\_Type, P.med\_report AS Medical\_History FROM Profiles P, Donors D, Collect C WHERE C.did = D.did AND D.did = @did AND P.SSN = @SSN", con);**

**cmd.Parameters.AddWithValue("@did", textBox\_donordid.Text);**

**cmd.Parameters.AddWithValue("@SSN", textBox\_donorSSN.Text);**

**SqlDataAdapter adapt = new SqlDataAdapter(cmd);**

**DataTable dt = new DataTable();**

**adapt.SelectCommand = cmd;**

**adapt.Fill(dt);**

**dataGridView1.DataSource = dt;**

**con.Close();**

**}**

**private void button\_login\_Click(object sender, EventArgs e)**

**{**

**if (textBox\_donordid.Text == "" || textBox\_donorSSN.Text == "")**

**{**

**MessageBox.Show("Please fill out all required fields!");**

**return;**

**}**

**show\_data();**

**}**

**private void textBox\_donordid\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**private void Donor\_Login\_Load(object sender, EventArgs e)**

**{**

**}**

**}**

**}**

**Patient Login Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**using System.Data.SqlClient;**

**namespace WindowsFormsApplication1**

**{**

**public partial class Patient\_Login : Form**

**{**

**SqlConnection con2 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd2;**

**public Patient\_Login()**

**{**

**InitializeComponent();**

**}**

**private void label1\_Click(object sender, EventArgs e)**

**{**

**}**

**private void show\_data()**

**{**

**con2.Open();**

**SqlCommand cmd2 = new SqlCommand("SELECT P.p\_name AS Name, P.addr AS Address, P.num AS Phone\_Number, P.b\_group AS Blood\_Type, P.med\_report AS Medical\_History FROM Profiles P, Patients X, Donate D WHERE X.pid = D.pid AND D.pid = @pid AND P.SSN = @SSN", con2);**

**cmd2.Parameters.AddWithValue("@pid", textBox\_patientpid.Text);**

**cmd2.Parameters.AddWithValue("@SSN", textBox\_patientssn.Text);**

**SqlDataAdapter adapt2 = new SqlDataAdapter(cmd2);**

**DataTable dt2 = new DataTable();**

**adapt2.SelectCommand = cmd2;**

**adapt2.Fill(dt2);**

**dataGridView3.DataSource = dt2;**

**con2.Close();**

**}**

**private void label1\_Click\_1(object sender, EventArgs e)**

**{**

**}**

**private void button\_login\_Click(object sender, EventArgs e)**

**{**

**if (textBox\_patientpid.Text == "" || textBox\_patientssn.Text == "")**

**{**

**MessageBox.Show("Please fill out all required fields!");**

**return;**

**}**

**show\_data();**

**}**

**private void textBox\_patientssn\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**private void textBox\_patientpid\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**}**

**}**

**Profile Delete Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**using System.Data.SqlClient;**

**namespace WindowsFormsApplication1**

**{**

**public partial class ProfileDelete : Form**

**{**

**SqlConnection con10 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd10;**

**public ProfileDelete()**

**{**

**InitializeComponent();**

**}**

**private void ClearData()**

**{**

**textBox\_name.Text = "";**

**}**

**private void button\_delete\_Click(object sender, EventArgs e)**

**{**

**if (textBox\_name.Text == "")**

**{**

**MessageBox.Show("Please input a valid name!");**

**}**

**else**

**{**

**try**

**{**

**cmd10 = new SqlCommand("DELETE FROM Profiles WHERE Profiles.p\_name = @name", con10);**

**con10.Open();**

**cmd10.Parameters.AddWithValue("@name", textBox\_name.Text);**

**cmd10.ExecuteNonQuery();**

**con10.Close();**

**MessageBox.Show("Record is deleted!");**

**ClearData();**

**}**

**catch (Exception ex)**

**{**

**MessageBox.Show(ex.Message);**

**con10.Close();**

**ClearData();**

**}**

**}**

**}**

**private void textBox\_name\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**}**

**}**

**Profile Insert Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**using System.Data.SqlClient;**

**namespace WindowsFormsApplication1**

**{**

**public partial class ProfileInsert : Form**

**{**

**SqlConnection con9 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd9;**

**public ProfileInsert()**

**{**

**InitializeComponent();**

**}**

**private void label1\_Click(object sender, EventArgs e)**

**{**

**}**

**private void ProfileInsert\_Load(object sender, EventArgs e)**

**{**

**}**

**private void ClearData()**

**{**

**textBox\_name.Text = "";**

**textBox\_ssn.Text = "";**

**textBox\_num.Text = "";**

**textBox\_addr.Text = "";**

**textBox\_b\_group.Text = "";**

**textBox\_med\_report.Text = "";**

**}**

**private void button\_insert\_Click(object sender, EventArgs e)**

**{**

**if (textBox\_name.Text == "" || textBox\_ssn.Text == "" || textBox\_num.Text == "" || textBox\_addr.Text == "" || textBox\_b\_group.Text == "")**

**{**

**MessageBox.Show("Please fill out all the fields!");**

**}**

**else**

**{**

**try**

**{**

**con9.Open();**

**SqlCommand cmd9 = new SqlCommand("INSERT INTO Profiles(Profiles.p\_name, Profiles.SSN, Profile.num, Profiles.addr, Profiles.b\_group, Profiles.med\_report) VALUES (@p\_name, @SSN, @num, @addr, @b\_group, @med\_report)", con9);**

**cmd9.Parameters.AddWithValue("@p\_name", textBox\_name.Text);**

**cmd9.Parameters.AddWithValue("@SSN", textBox\_ssn.Text);**

**cmd9.Parameters.AddWithValue("@num", textBox\_num.Text);**

**cmd9.Parameters.AddWithValue("@addr", textBox\_addr.Text);**

**cmd9.Parameters.AddWithValue("@b\_group", textBox\_b\_group.Text);**

**cmd9.Parameters.AddWithValue("@med\_report", textBox\_med\_report.Text);**

**cmd9.ExecuteNonQuery();**

**con9.Close();**

**MessageBox.Show("Profile Inserted Successfully");**

**ClearData();**

**}**

**catch (Exception ex)**

**{**

**MessageBox.Show(ex.Message);**

**con9.Close();**

**ClearData();**

**}**

**}**

**}**

**private void textBox\_ssn\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**private void textBox\_name\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**}**

**}**

**Profile Update Code:**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**using System.Data.SqlClient;**

**namespace WindowsFormsApplication1**

**{**

**public partial class ProfileUpdate : Form**

**{**

**SqlConnection con11 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integrated Security=True");**

**SqlCommand cmd11;**

**public ProfileUpdate()**

**{**

**InitializeComponent();**

**}**

**private void ClearData()**

**{**

**textBox\_name.Text = "";**

**textBox\_updated\_med\_report.Text = "";**

**}**

**private void button\_update\_Click(object sender, EventArgs e)**

**{**

**if (textBox\_name.Text == "")**

**{**

**MessageBox.Show("Please fill out all the required fields!");**

**}**

**try**

**{**

**con11.Open();**

**SqlCommand cmd11 = new SqlCommand("UPDATE Profiles SET Profiles.med\_report = @med\_report WHERE Profiles.p\_name = @p\_name", con11);**

**cmd11.Parameters.AddWithValue("@med\_report", textBox\_updated\_med\_report.Text);**

**cmd11.Parameters.AddWithValue("@p\_name", textBox\_name.Text);**

**cmd11.ExecuteNonQuery();**

**con11.Close();**

**MessageBox.Show("Profile has been updated!");**

**ClearData();**

**}**

**catch (Exception ex)**

**{**

**MessageBox.Show(ex.Message);**

**con11.Close();**

**ClearData();**

**}**

**}**

**private void label\_name\_Click(object sender, EventArgs e)**

**{**

**}**

**private void textBox\_name\_TextChanged(object sender, EventArgs e)**

**{**

**}**

**}**

**}**

**1.3 CONCLUSION AND WORKS CITED**

**1.3.1 Conclusion**

In order to effectively complete this project, we were forced to alter our original database from Project 1. We removed interchangeable attributes in order to prevent repetition, and we connected patients and donors as children to the greater entity called profile. By doing so, we were able to create the database in BCNF/3NF form through relationships between our functional dependencies. Once we concretely changed our database format, we were able to create the tables and queries in Visual Studio using our new entities/keys/relationships/attributes. The alteration design allowed for a more efficient presentation of data. We created our triggers and views randomly in accordance to what people would want to see. Our queries were derived from our first project before alterations were made to our database, thus, the variables/relationships/keys/entities are presented differently than they are in our final project.

We created the GUI from every perspective: Patients, Donors, and Blood Banks. Our GUI Design begins with a Blood Bank Profile window which has three options; Donor, Patient, and Blood Bank. By clicking on Donor, you open a new window which asks for you to input your Donor ID and SSN. This is what we consider our “Login” page. The Donor ID is viewed as a username and the SSN is viewed as a password. By entering their username and password, the Donors profile information is displayed on the screen. If they enter an invalid Donor ID, an error message prints to the screen. Similarly, by clicking the Patient button on the Blood Bank Profile, you are directed to a similar window, but in this window, the Patient ID is the username and the SSN is still the password. By entering in a valid Patient ID and SSN, the Patient’s profile information is printed to the screen. Invalid Patient ID’s display an error message. Finally, we have a Blood Bank button. Clicking this button redirects you to a new window which has a few buttons/options. You are permitted to search a blood bank’s name, and by doing so, you are shown the phone number and address of the specific blood bank. Another option you have is to insert a new profile. By clicking the insert new profile button, you are given another window. The window asks you to input all the necessary information needed for a new profile; Name, SSN, Phone Number, Address, Blood Group, and Medical History. You are not permitted to leave any of the information blank except for the medical history. If you leave one of the required fields blank, an error message is prompted. If you successfully input all of your data, a message telling you that you successfully inserted a profile is prompted. Similarly, we have a Delete Profile button. By clicking the delete profile button, you are redirected to a new window. This window prompts you to enter your name. If you enter an invalid name, an error messages is prompted, otherwise, you are notified by a message telling you that you have successfully deleted the profile of the name you entered. The Update Profile button prompts a new window. The window asks you to input your name, which is mandatory, if you fail to do so, an error message is prompted. The update profile is used to update the patient/donors medical history**.** You are permitted to leave the medical history textbox blank as this would mean that you would like to clear your history.At the bottom of the page, we have two more buttons. The Sort Profiles button sorts all of the profiles within the profile table alphabetically in ascending order. The Count Blood Type button displays the different blood types within the profile and how often they are shown. This is our interpretation for avg/min/max for our data set since we do not have any numerical values that need to be/should be averaged, min(ed), or max(ed).

For future work, we would like to be able to average the occurrences of different blood types along with displaying the maximum (most common) blood type and the minimum (least common) blood type within our database. Similarly, we would like to group the Patients/Donors in accordance to the specific blood bank they are apart of because when trying to implement this in our current project, we ran into an abundance of errors and issues such as repetition, incorrect matching, etc. Overall, in the future, we would like to make separate profiles for individual test banks and do the specific aggregations (sort, avg, etc.) for specific Blood Banks.

**WORKS CITED**

Ramakrishanan, R. & Gehrke, J. (200). *Database Management System*. Boston, MA: McGraw-Hill.