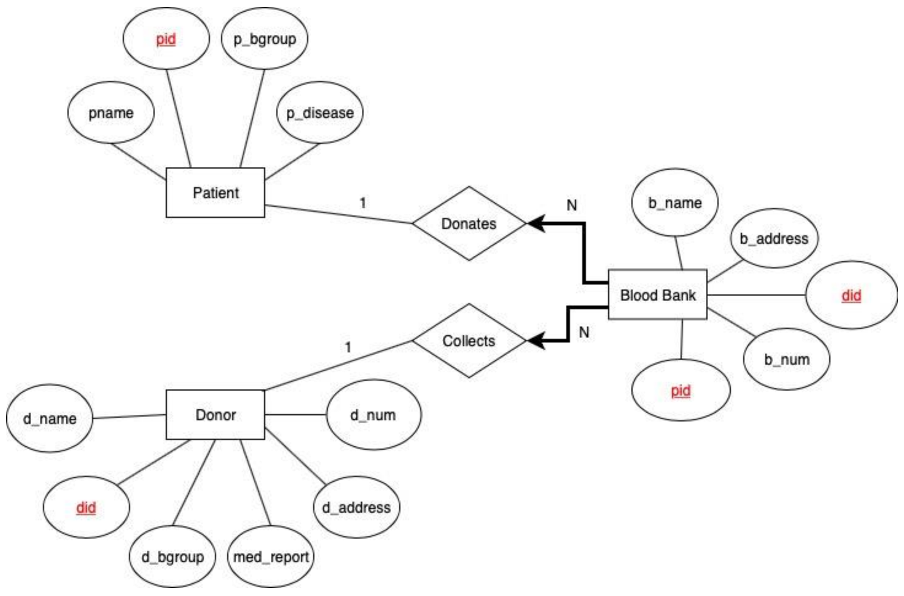
Karen Mani & Sasha Smolyansky
Professor He
CSCI 2441W - Project 2
27 April 2019

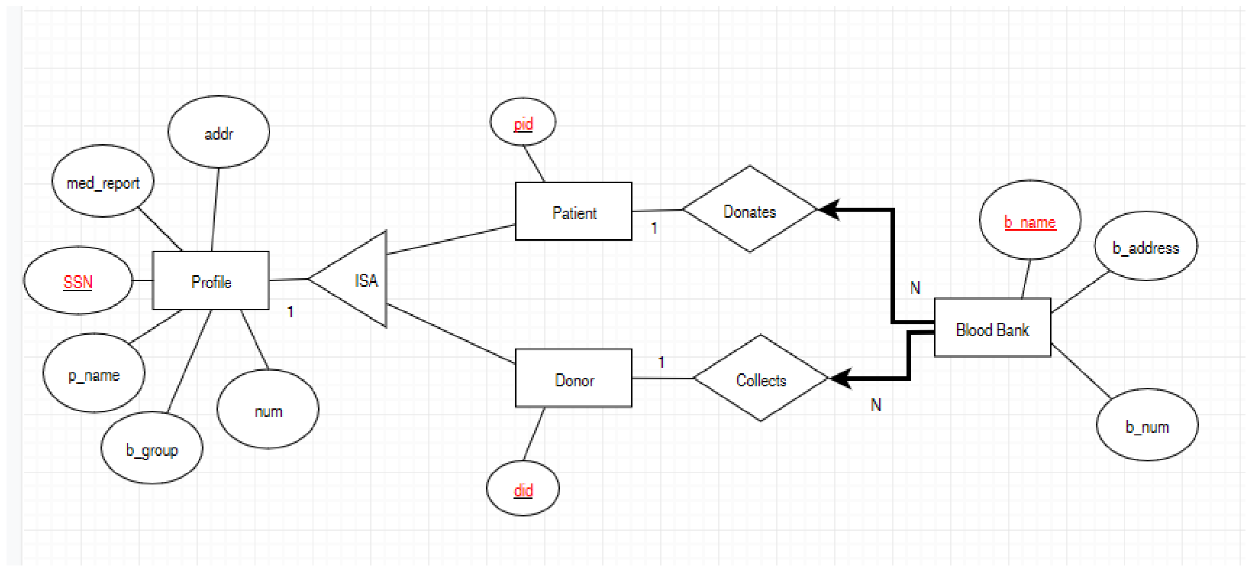
Project 2 Report

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(<u>*SSN*</u>, 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(<u>*did*</u>, <u>*b_name*</u>): No FDs, in BCNF Form

Collects(<u>*pid*</u>, <u>*b_name*</u>): No FDs, in BCNF Form

1.2 DATABASE IMPLEMENTATION AND GUI DESIGN

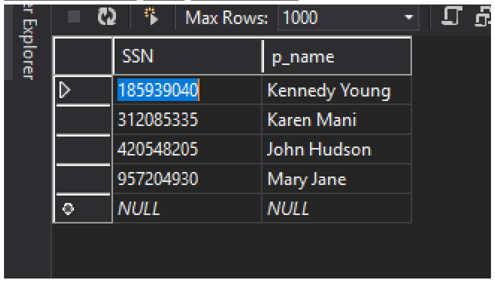
1.2.1 Creating Tables and Inputting Data

```
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')
```

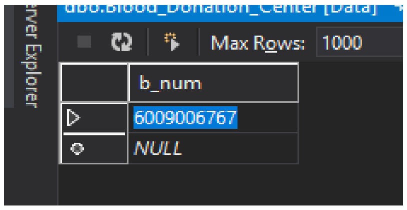
1.2.2 Views NEW

```
CREATE VIEW AB_positive
AS SELECT SSN,p_name
FROM Profiles
WHERE b_group = 'AB+';
```



SSN	p_name
185939040	Kennedy Young
312085335	Karen Mani
420548205	John Hudson
957204930	Mary Jane
NULL	NULL

```
CREATE VIEW Blood_Donation_Center
AS SELECT b_num
FROM BloodBanks
WHERE b_name = 'Blood Donation Center';
```


dbo.Blood_Donation_Center [Data]

Max Rows: 1000

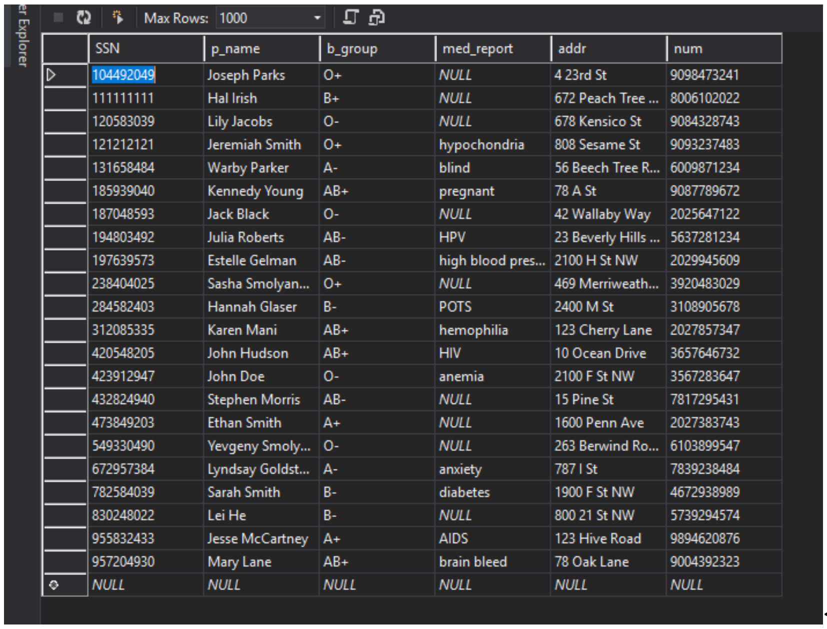
	b_num
▷	6009006767
	NULL

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';
```

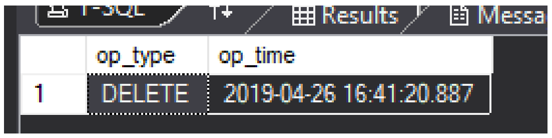

Max Rows: 1000

	SSN	p_name	b_group	med_report	addr	num
▷	104492049	Joseph Parks	O+	NULL	4 23rd St	9098473241
	111111111	Hal Irish	B+	NULL	672 Peach Tree ...	8006102022
	120583039	Lily Jacobs	O-	NULL	678 Kensico St	9084328743
	121212121	Jeremiah Smith	O+	hypochondria	808 Sesame St	9093237483
	131658484	Warby Parker	A-	blind	56 Beech Tree R...	6009871234
	185939040	Kennedy Young	AB+	pregnant	78 A St	9087789672
	187048593	Jack Black	O-	NULL	42 Wallaby Way	2025647122
	194803492	Julia Roberts	AB-	HPV	23 Beverly Hills ...	5637281234
	197639573	Estelle Gelman	AB-	high blood pres...	2100 H St NW	2029945609
	238404025	Sasha Smolyan...	O+	NULL	469 Merriweath...	3920483029
	284582403	Hannah Glaser	B-	POTS	2400 M St	3108905678
	312085335	Karen Mani	AB+	hemophilia	123 Cherry Lane	2027857347
	420548205	John Hudson	AB+	HIV	10 Ocean Drive	3657646732
	423912947	John Doe	O-	anemia	2100 F St NW	3567283647
	432824940	Stephen Morris	AB-	NULL	15 Pine St	7817295431
	473849203	Ethan Smith	A+	NULL	1600 Penn Ave	2027383743
	549330490	Yevgeny Smoly...	O-	NULL	263 Berwind Ro...	6103899547
	672957384	Lyndsay Goldst...	A-	anxiety	787 I St	7839238484
	782584039	Sarah Smith	B-	diabetes	1900 F St NW	4672938989
	830248022	Lei He	B-	NULL	800 21 St NW	5739294574
	955832433	Jesse McCartney	A+	AIDS	123 Hive Road	9894620876
	957204930	Mary Lane	AB+	brain bleed	78 Oak Lane	9004392323
	NULL	NULL	NULL	NULL	NULL	NULL

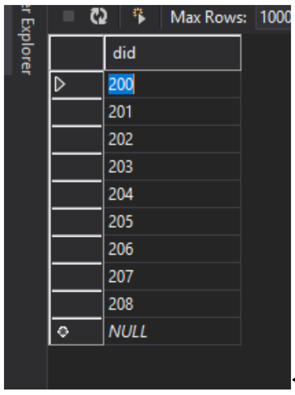
<-- last line

```
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;
```



	op_type	op_time
1	DELETE	2019-04-26 16:41:20.887



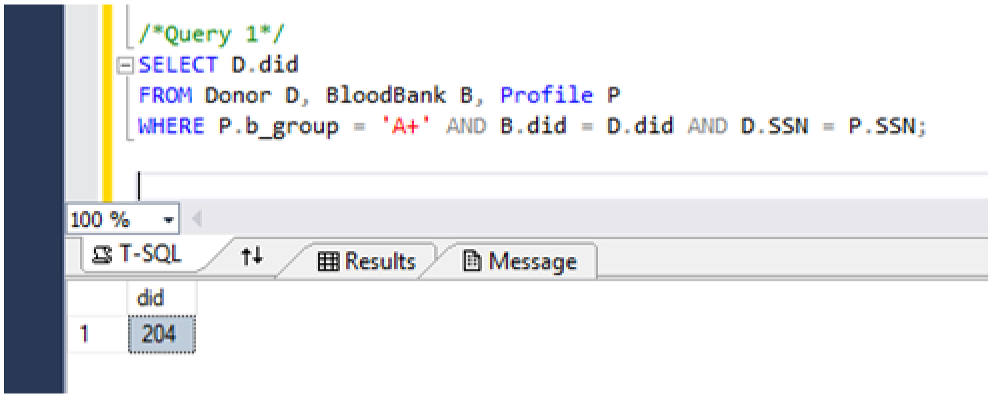
```
did
200
201
202
203
204
205
206
207
208
NULL
```

<-- 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.

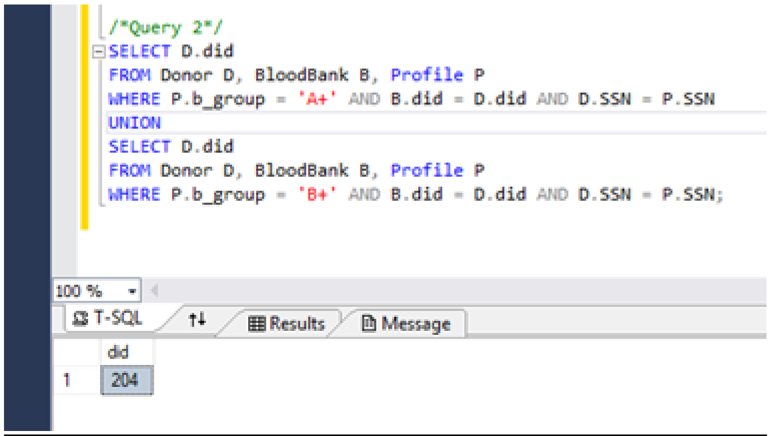
1.
   ```
   SELECT D.did
   FROM Donor D, BloodBank B, Profile P
   WHERE P.b_group = 'A+' AND B.did = D.did AND D.SSN = P.SSN;
   ```

   ```
   /*Query 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;
   ```

   ```
   100 %
   T-SQL   Results   Message
     did
   1 204
   ```

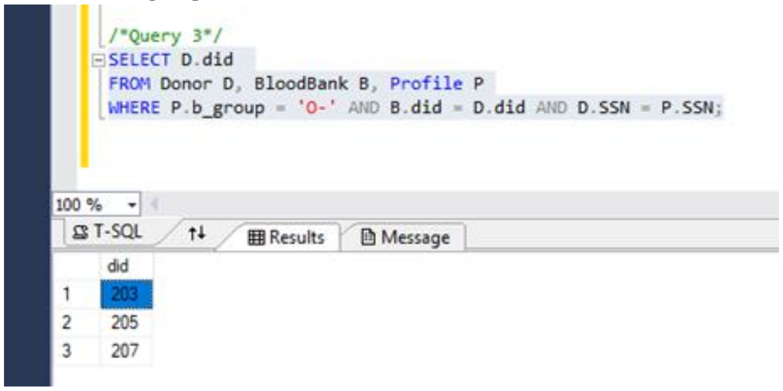
2.
   ```
   SELECT D.did
   FROM Donor D, BloodBank B, Profile P
   WHERE P.b_group = 'A+' AND B.did = D.did AND D.SSN = P.SSN
   UNION
   SELECT D.did
   FROM Donor D, BloodBank B, Profile P
   WHERE P.b_group = 'B+' AND B.did = D.did AND D.SSN = P.SSN;
   ```



```
/*Query 2*/
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;
```

```
   did
1  204
```

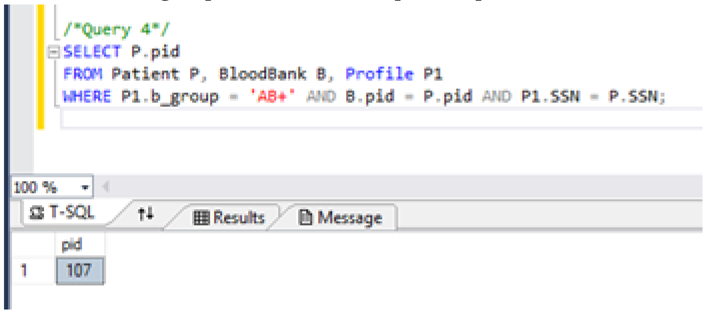
3. SELECT D.did
   FROM Donor D, BloodBank B, Profile P
   WHERE P.b_group = 'O-' AND B.did = D.did AND D.SSN = P.SSN;



```
/*Query 3*/
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;
```

```
   did
1  203
2  205
3  207
```

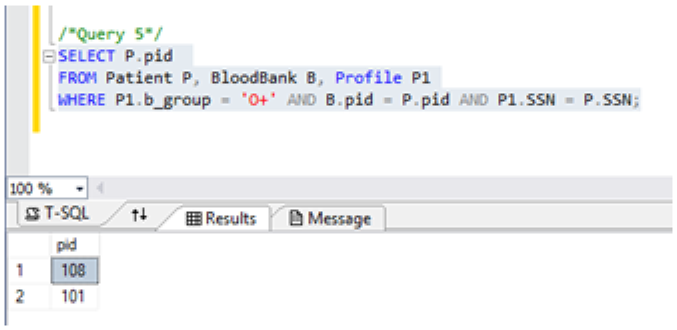
4. SELECT P.pid
   FROM Patient P, BloodBank B, Profile P1
   WHERE P1.b_group = 'AB+' AND B.pid = P.pid AND P1.SSN = P.SSN;



```
/*Query 4*/
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;
```

```
   pid
1  107
```

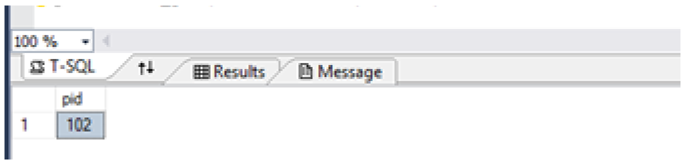
5. SELECT P.pid
   FROM Patient P, BloodBank B, Profile P1
   WHERE P1.b_group = 'O+' AND B.pid = P.pid AND P1.SSN = P.SSN;



```
/*Query 5*/
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;
```

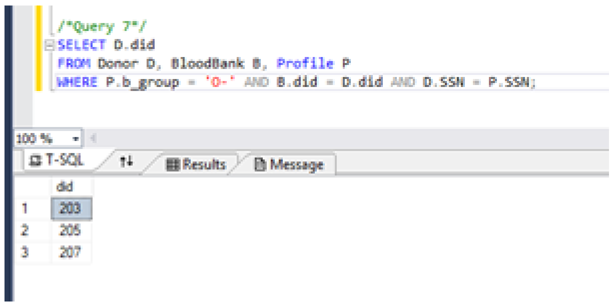
```
   pid
1  108
2  101
```

6. 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



```
   pid
1  102
```

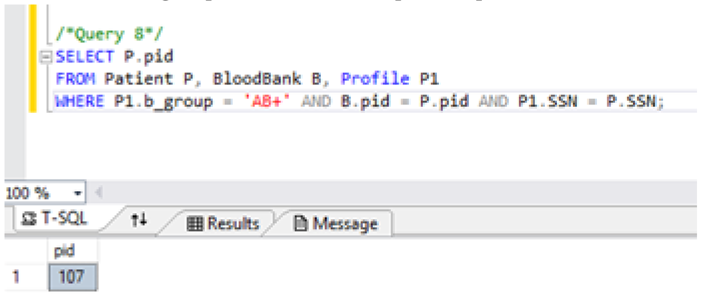
7. SELECT D.did
   FROM Donor D, BloodBank B, Profile P
   WHERE P.b_group = 'O-' AND B.did = D.did AND D.SSN = P.SSN;



```
/*Query 7*/
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;
```

```
   did
1  203
2  205
3  207
```

8. 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;

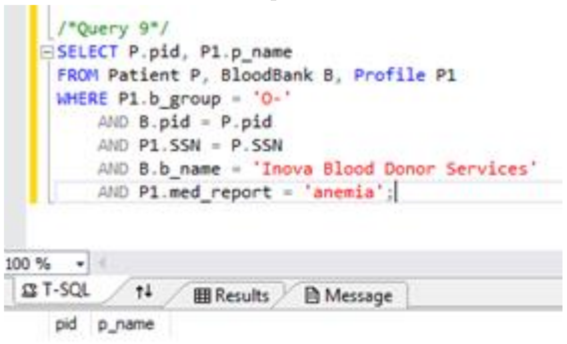


```
/*Query 8*/
SELECT P.pid
FROM Patient P, BloodBank B, Profile P1
WHERE P1.b_group = 'AB+' AND B.pid = P.pid AND P1.SSN = P.SSN;
```

```
   pid
1  107
```

9. 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

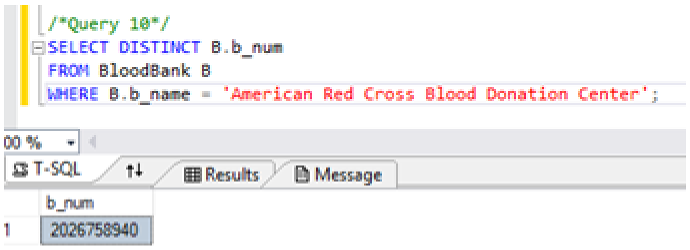
```
          AND B.b_name = 'Inova Blood Donor Services'
          AND P1.med_report = 'anemia';
```



```
/*Query 9*/
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';
```

```
100 %
T-SQL    Results    Message
pid   p_name
```

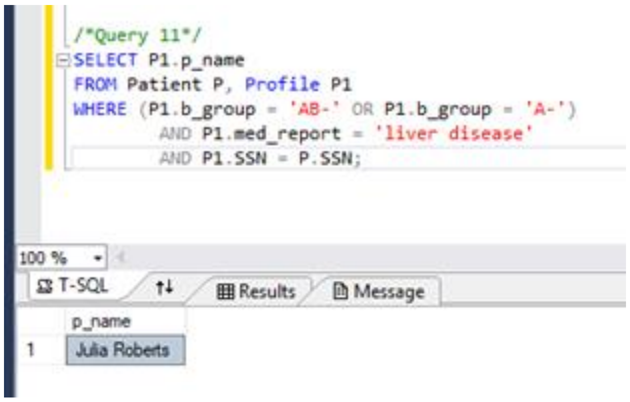
10.
```
SELECT DISTINCT B.b_num
FROM BloodBank B
WHERE B.b_name = 'American Red Cross Blood Donation Center';
```



```
/*Query 10*/
SELECT DISTINCT B.b_num
FROM BloodBank B
WHERE B.b_name = 'American Red Cross Blood Donation Center';
```

```
00 %
T-SQL    Results    Message
   b_num
1  2026758940
```

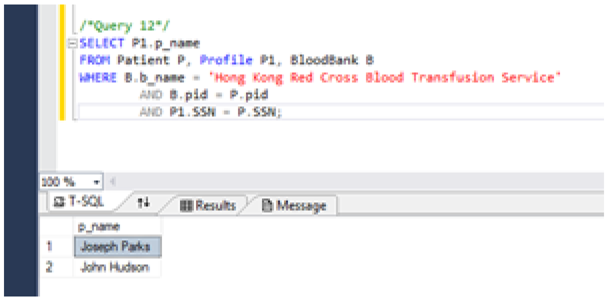
11.
```
SELECT P1.p_name
FROM Patient P, Profile P1
WHERE (P1.b_group = 'AB-' OR P1.b_group = 'A-')
          AND P1.med_report = 'liver disease'
          AND P1.SSN = P.SSN;
```



```
/*Query 11*/
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;
```

```
100 %
T-SQL    Results    Message
   p_name
1  Julia Roberts
```

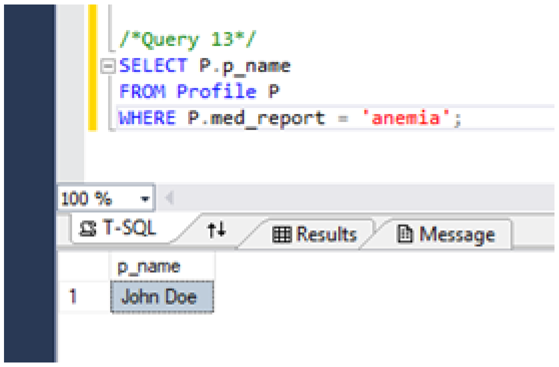
12.
```
SELECT P1.p_name
FROM Patient P, Profile P1, BloodBank B
WHERE B.b_name = 'Hong Kong Red Cross Blood Transfusion Service'
          AND B.pid = P.pid
          AND P1.SSN = P.SSN;
```



```
/*Query 12*/
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;
```

```
   p_name
1  Joseph Parks
2  John Hudson
```

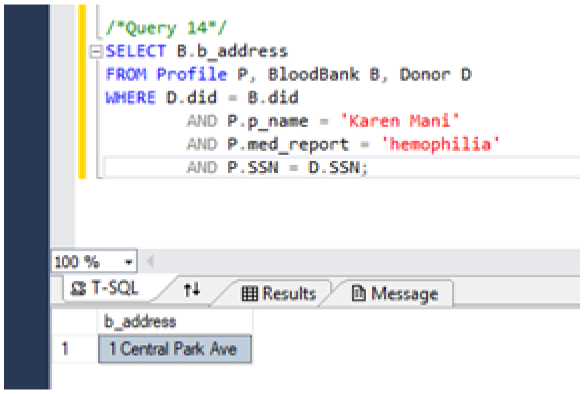
13. SELECT P.p_name
FROM Profile P
WHERE P.med_report = 'anemia';



```
/*Query 13*/
SELECT P.p_name
FROM Profile P
WHERE P.med_report = 'anemia';
```

```
   p_name
1  John Doe
```

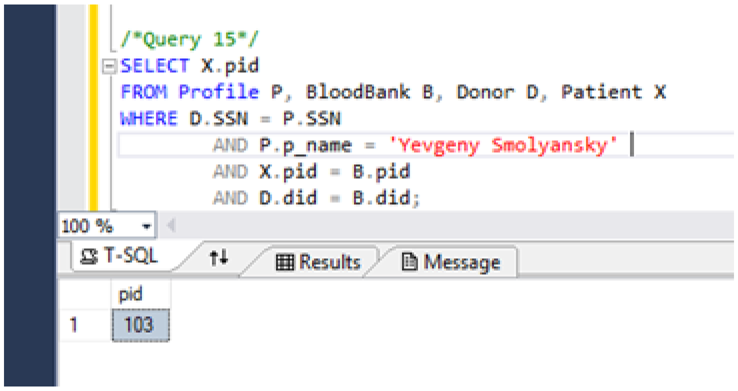
14. SELECT B.b_address
FROM Profile P, BloodBank B, Donor D
WHERE D.did = B.did
AND P.p_name = 'Karen Mani'
AND P.med_report = 'hemophilia'
AND P.SSN = D.SSN;



```
/*Query 14*/
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;
```

```
   b_address
1  1 Central Park Ave
```

15. 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;

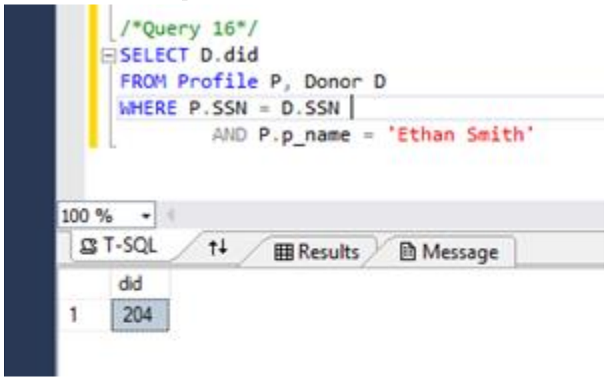


```
/*Query 15*/
SELECT X.pid
FROM Profile P, BloodBank B, Donor D, Patient X
WHERE D.SSN = P.SSN
        AND P.p_name = 'Yevgeny Smolyansky'
        AND X.pid = B.pid
        AND D.did = B.did;
```

```
     pid
1    103
```

16. SELECT D.did

```
FROM Profile P, Donor D
WHERE P.SSN = D.SSN
        AND P.p_name = 'Ethan Smith'
```

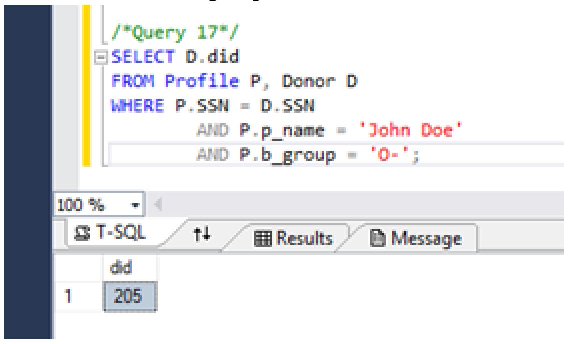


```
/*Query 16*/
SELECT D.did
FROM Profile P, Donor D
WHERE P.SSN = D.SSN
        AND P.p_name = 'Ethan Smith'
```

```
     did
1    204
```

17. SELECT D.did

```
FROM Profile P, Donor D
WHERE P.SSN = D.SSN
        AND P.p_name = 'John Doe'
        AND P.b_group = 'O-';
```

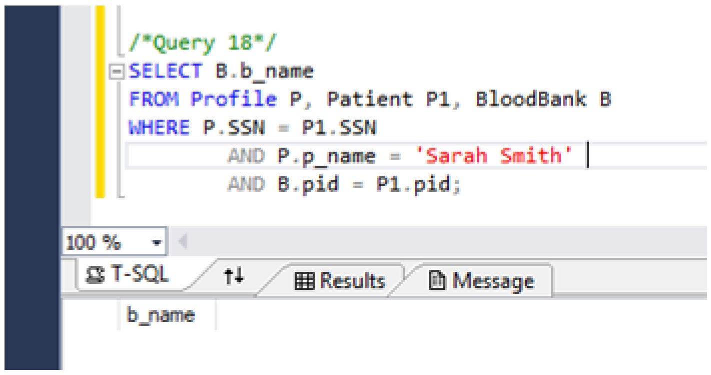


```
/*Query 17*/
SELECT D.did
FROM Profile P, Donor D
WHERE P.SSN = D.SSN
        AND P.p_name = 'John Doe'
        AND P.b_group = 'O-';
```

```
     did
1    205
```

18. SELECT B.b_name

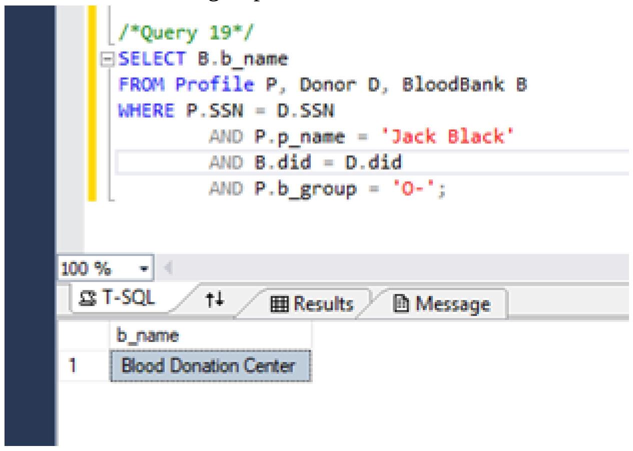
```
FROM Profile P, Patient P1, BloodBank B
WHERE P.SSN = P1.SSN
        AND P.p_name = 'Sarah Smith'
        AND B.pid = P1.pid;
```



```
/*Query 18*/
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;
```

```
b_name
```

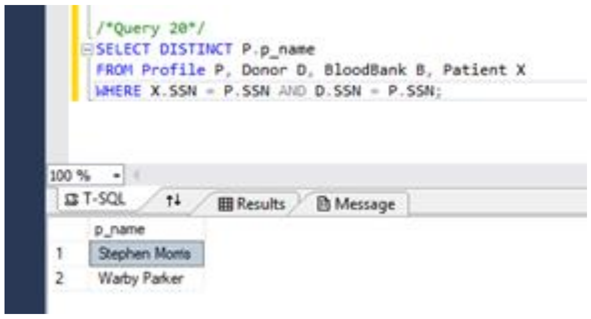
19. SELECT B.b_name
    FROM Profile P, Donor D, BloodBank B
    WHERE P.SSN = D.SSN
        AND P.p_name = 'Jack Black'
        AND B.did = D.did
        AND P.b_group = 'O-';



```
/*Query 19*/
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-';
```

```
   b_name
1  Blood Donation Center
```

20. SELECT DISTINCT P.p_name
    FROM Profile P, Donor D, BloodBank B, Patient X
    WHERE X.SSN = P.SSN AND D.SSN = P.SSN;



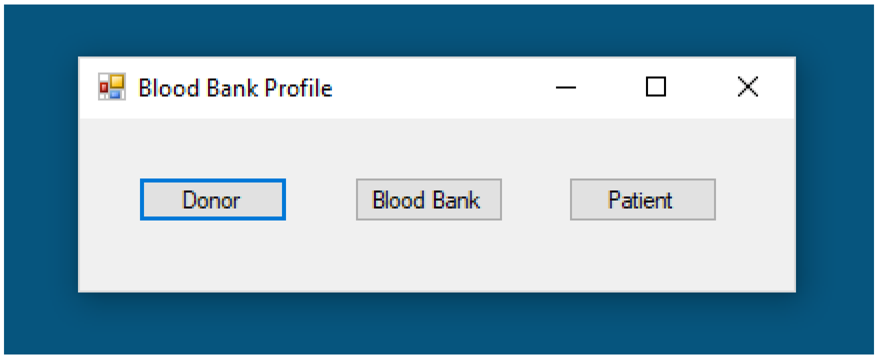
```
/*Query 20*/
SELECT DISTINCT P.p_name
FROM Profile P, Donor D, BloodBank B, Patient X
WHERE X.SSN = P.SSN AND D.SSN = P.SSN;
```

```
   p_name
1  Stephen Morris
2  Warby Parker
```

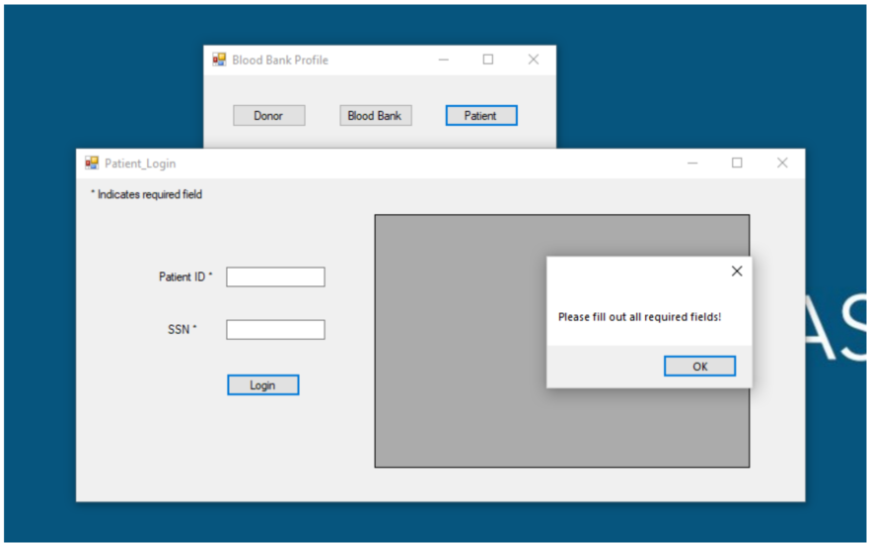
1.2.5 GUI Design

1.2.5.1 Images of the Database

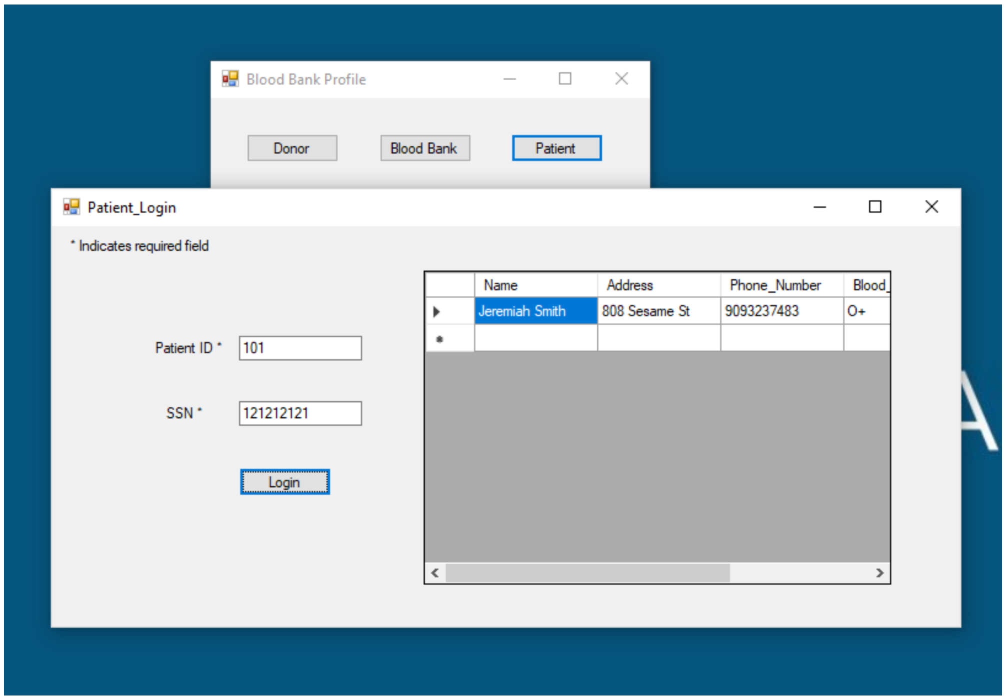
Homepage



Patient Login



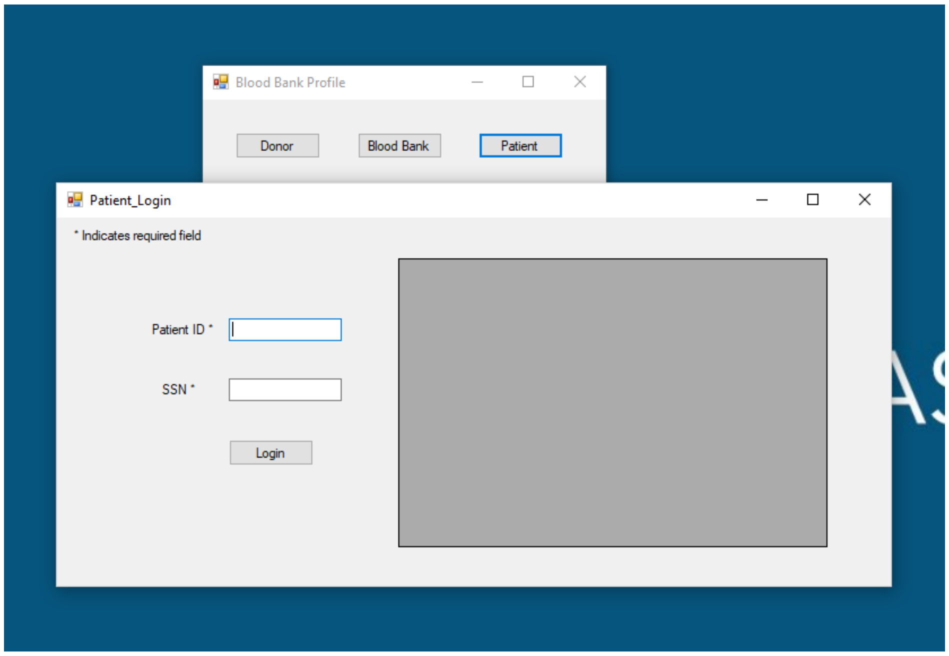
Please fill out all required fields!



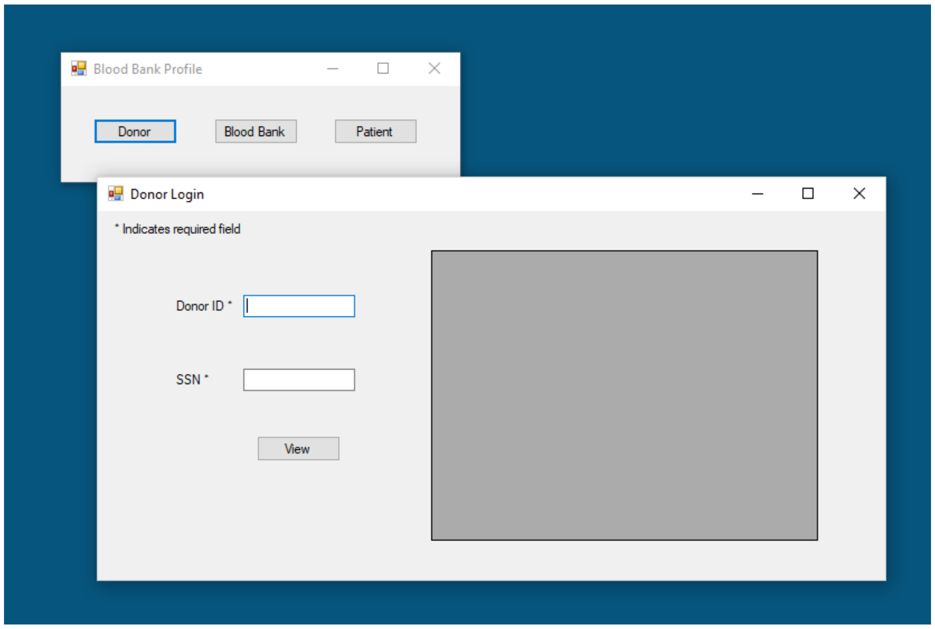
Patient ID * 101

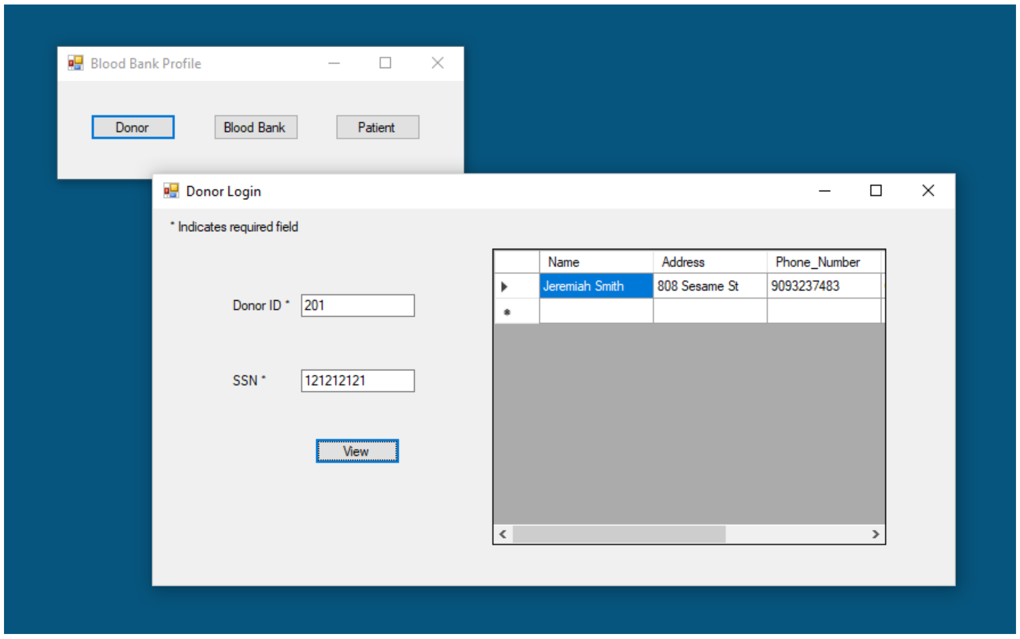
SSN * 121212121

Name	Address	Phone_Number	Blood_
Jeremiah Smith	808 Sesame St	9093237483	O+


Blood Bank Profile
Donor
Blood Bank
Patient
Patient_Login
* Indicates required field
Patient ID *
SSN *
Login

Donor Login


Blood Bank Profile
Donor
Blood Bank
Patient
Donor Login
* Indicates required field
Donor ID *
SSN *
View

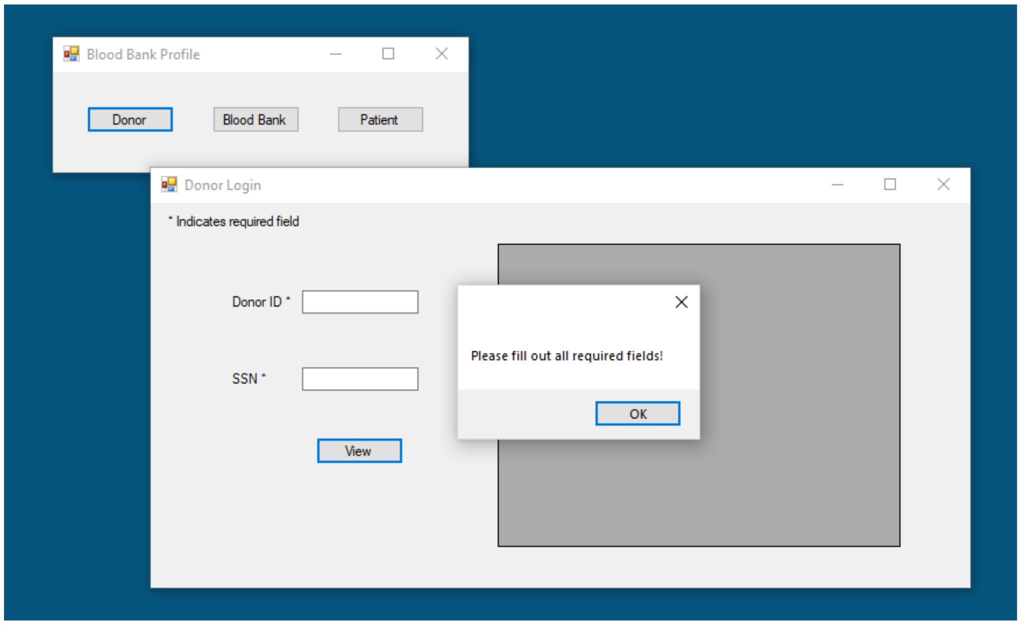


* Indicates required field

Donor ID * 201

SSN * 121212121

	Name	Address	Phone_Number
▶	Jeremiah Smith	808 Sesame St	9093237483
*			



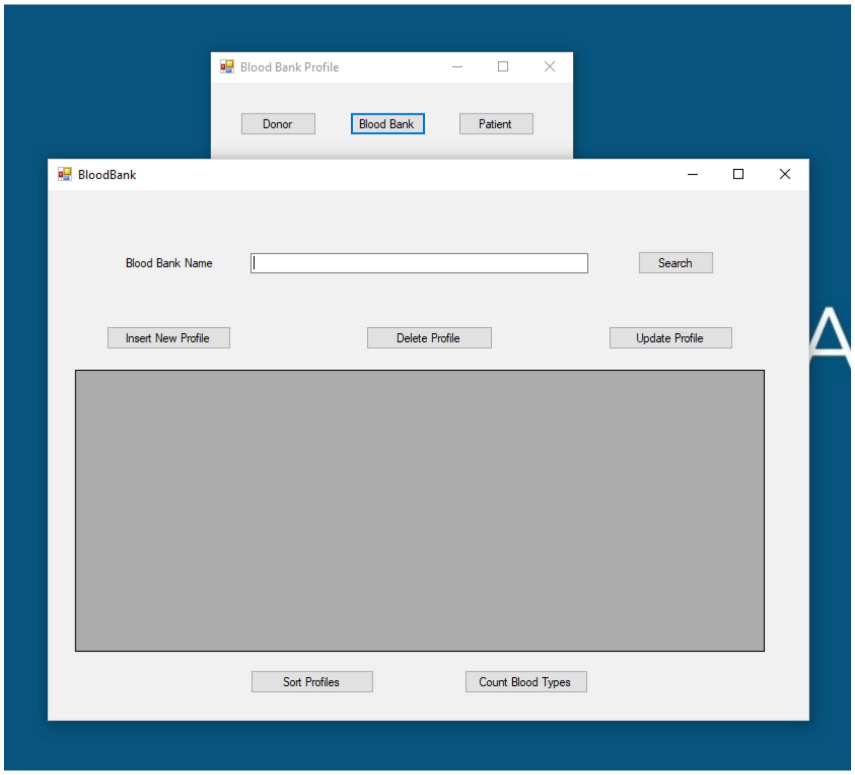
* Indicates required field

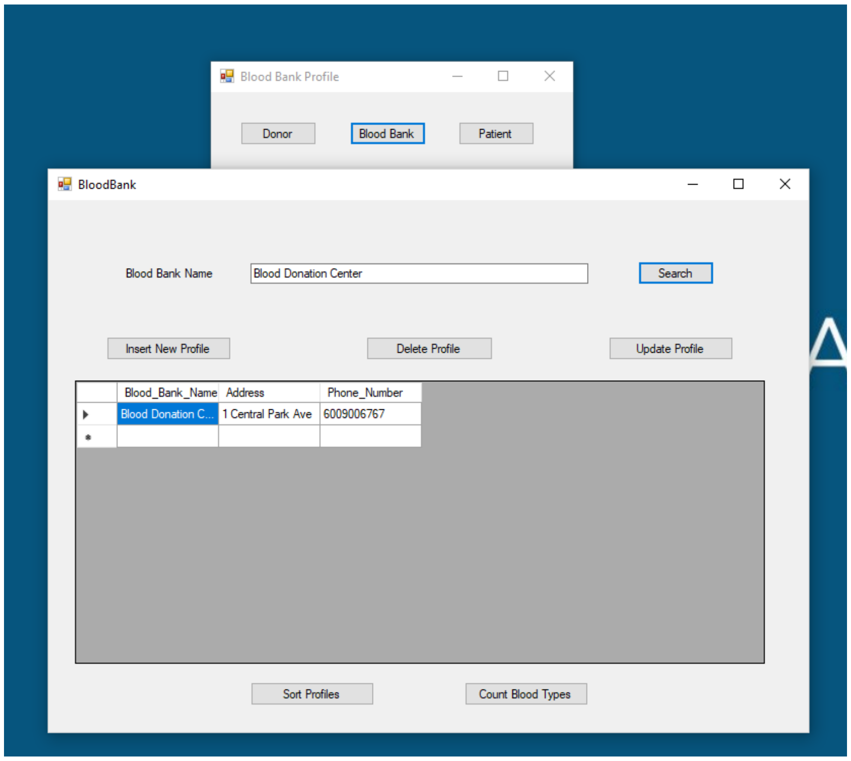
Donor ID *

SSN *

Please fill out all required fields!

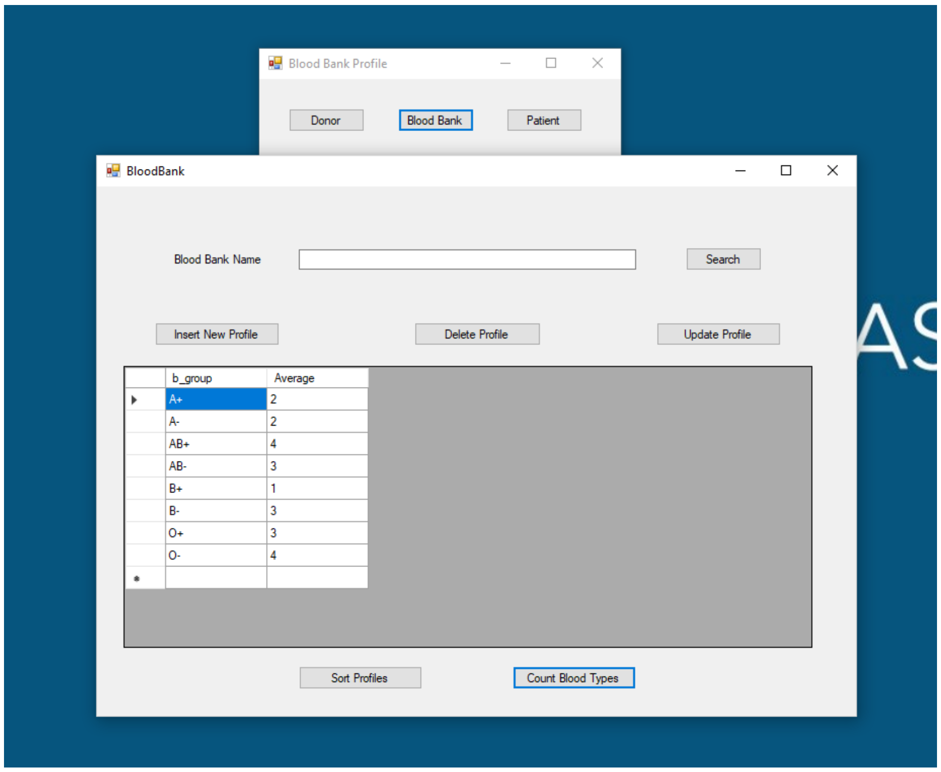
Blood Bank



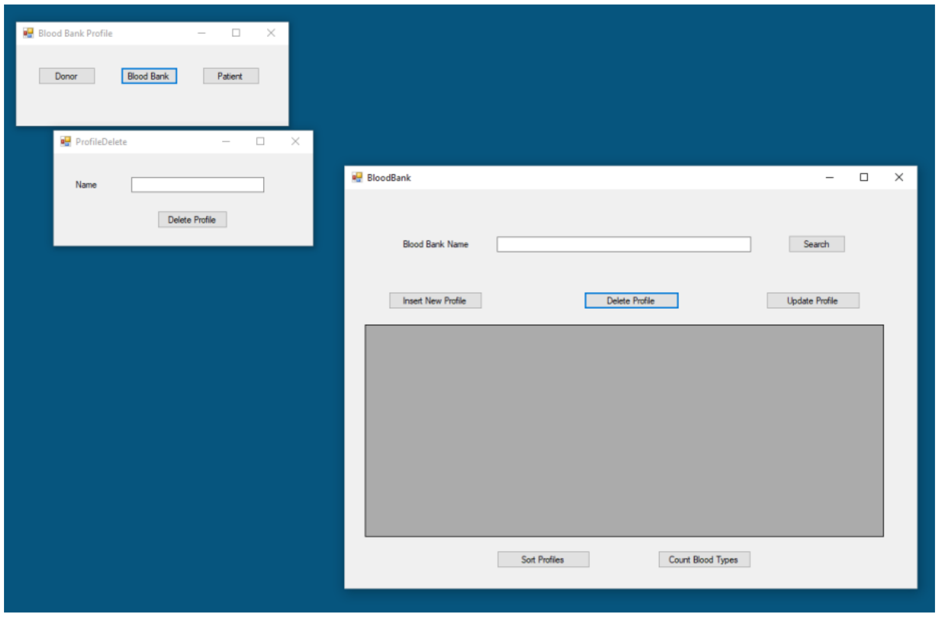


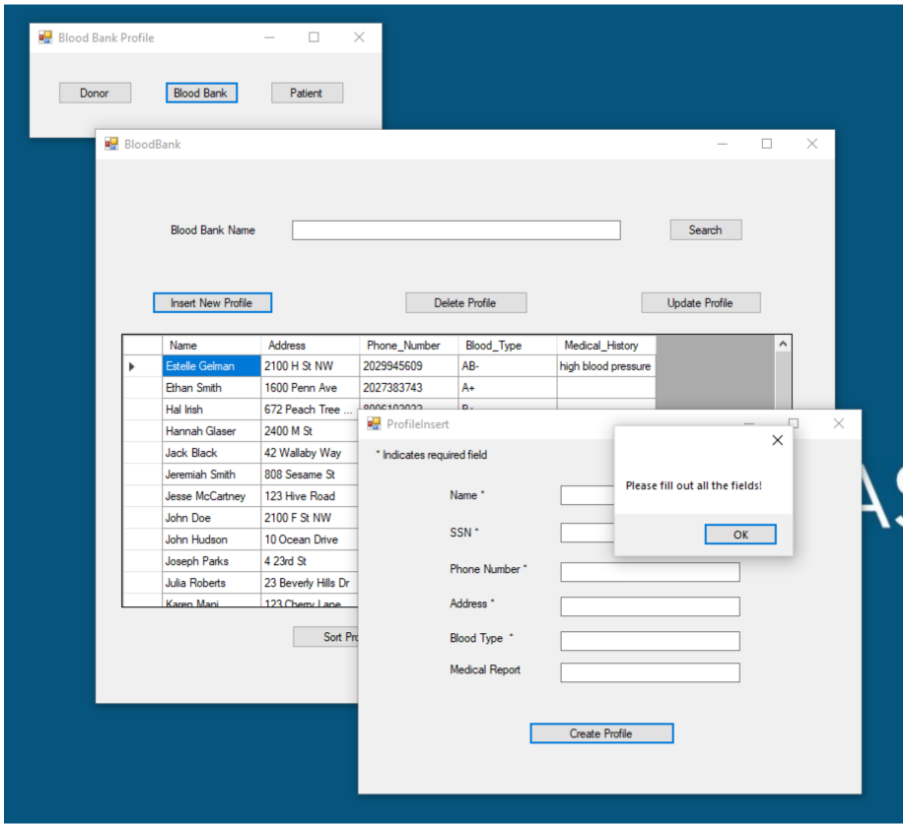
Blood Bank Name: Blood Donation Center

Blood_Bank_Name	Address	Phone_Number
Blood Donation C...	1 Central Park Ave	6009006767



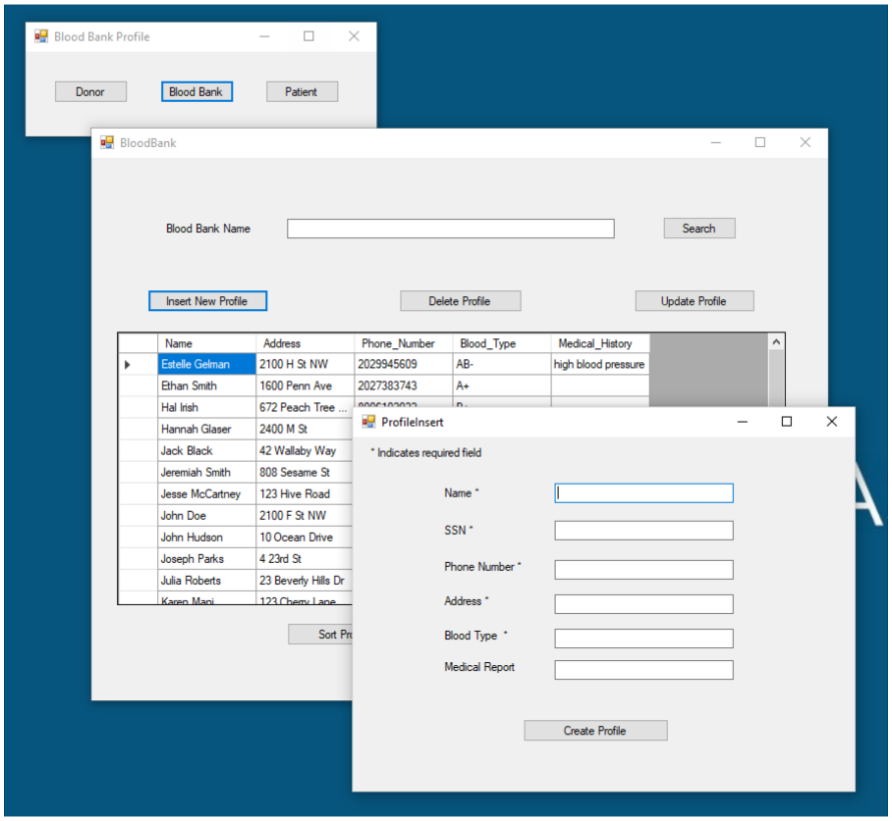
b_group	Average
A+	2
A-	2
AB+	4
AB-	3
B+	1
B-	3
O+	3
O-	4



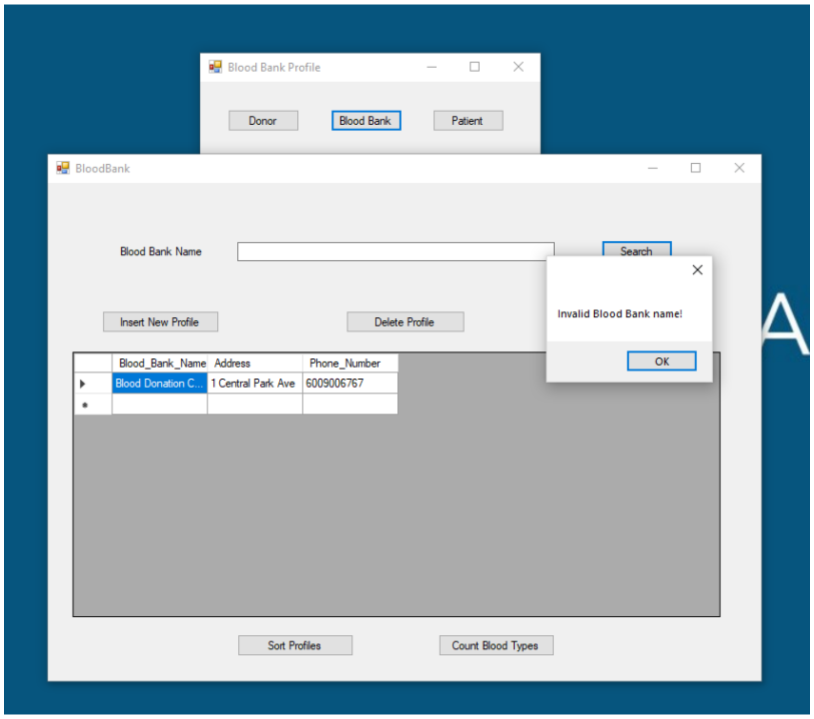


	Name	Address	Phone_Number	Blood_Type	Medical_History
▸	Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
	Ethan Smith	1600 Penn Ave	2027383743	A+	
	Hal Irish	672 Peach Tree ...			
	Hannah Glaser	2400 M St			
	Jack Black	42 Wallaby Way			
	Jeremiah Smith	808 Sesame St			
	Jesse McCartney	123 Hive Road			
	John Doe	2100 F St NW			
	John Hudson	10 Ocean Drive			
	Joseph Parks	4 23rd St			
	Julia Roberts	23 Beverly Hills Dr			
	Karen Mani	123 Cherry Lane			

Please fill out all the fields!



	Name	Address	Phone_Number	Blood_Type	Medical_History
▸	Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
	Ethan Smith	1600 Penn Ave	2027383743	A+	
	Hal Irish	672 Peach Tree ...			
	Hannah Glaser	2400 M St			
	Jack Black	42 Wallaby Way			
	Jeremiah Smith	808 Sesame St			
	Jesse McCartney	123 Hive Road			
	John Doe	2100 F St NW			
	John Hudson	10 Ocean Drive			
	Joseph Parks	4 23rd St			
	Julia Roberts	23 Beverly Hills Dr			
	Karen Mani	123 Cherry Lane			



Blood Bank Profile

Donor | Blood Bank | Patient

BloodBank

Blood Bank Name | Search

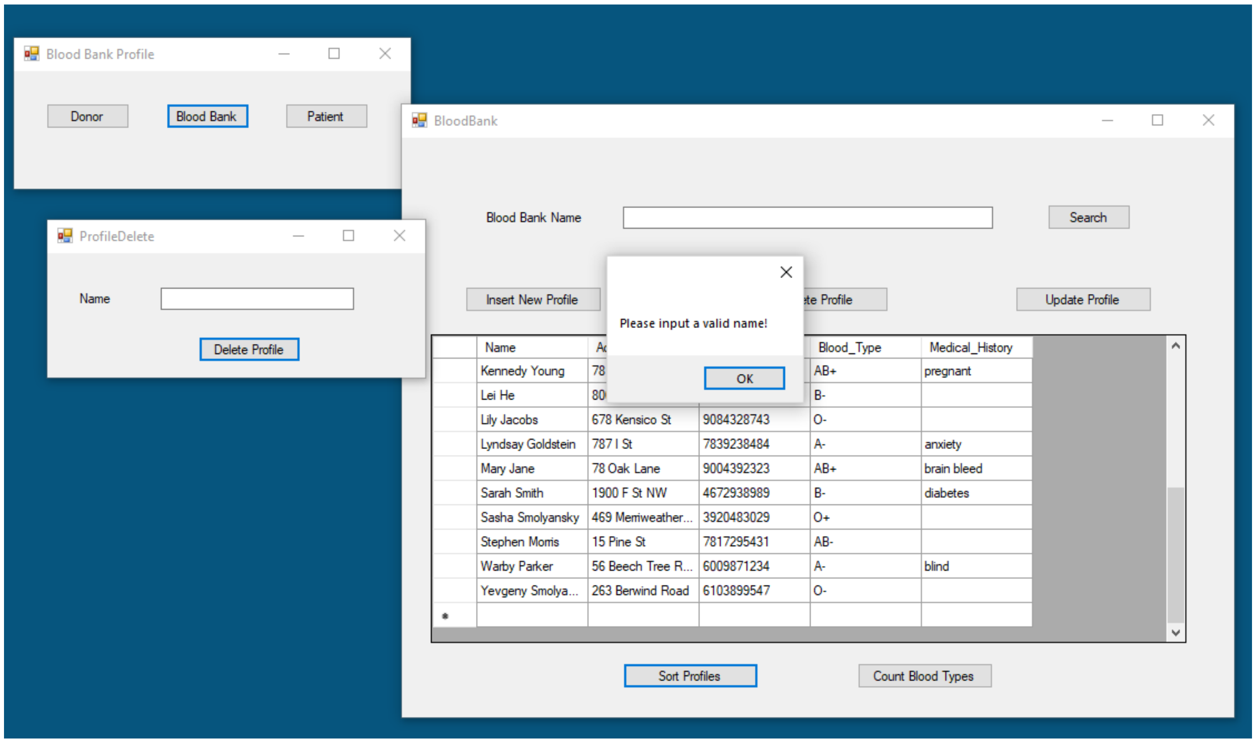
Insert New Profile | Delete Profile

Invalid Blood Bank name!

OK

	Blood_Bank_Name	Address	Phone_Number
▸	Blood Donation C...	1 Central Park Ave	6009006767
*			

Sort Profiles | Count Blood Types



Blood Bank Profile

Donor | Blood Bank | Patient

ProfileDelete

Name

Delete Profile

BloodBank

Blood Bank Name | Search

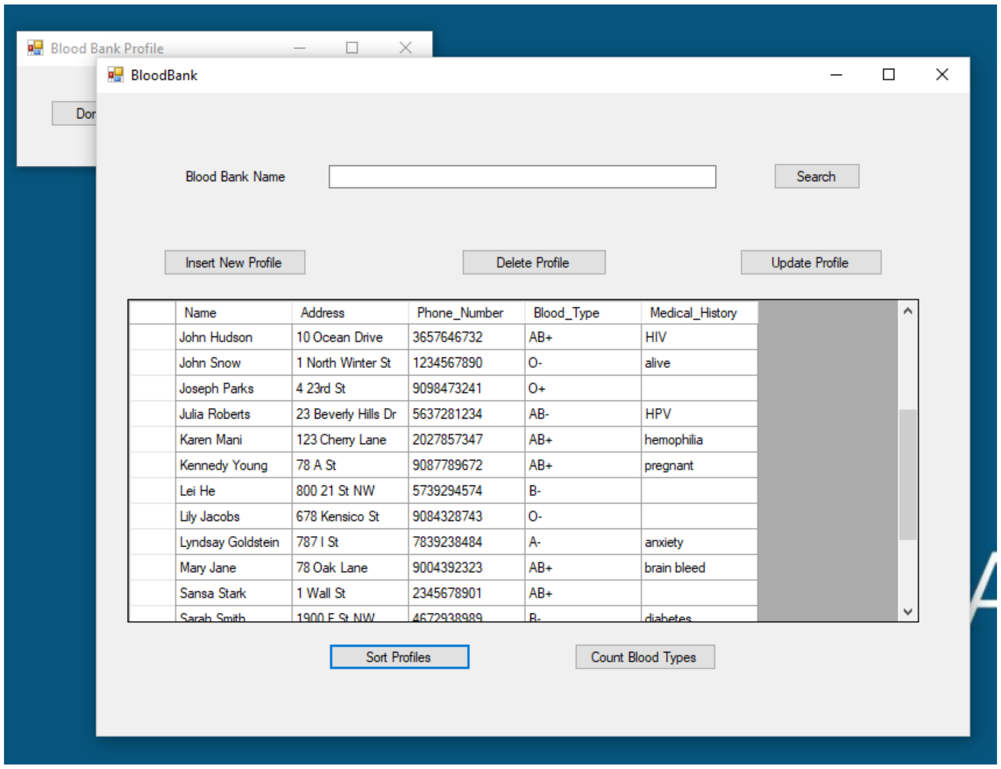
Insert New Profile | te Profile | Update Profile

Please input a valid name!

OK

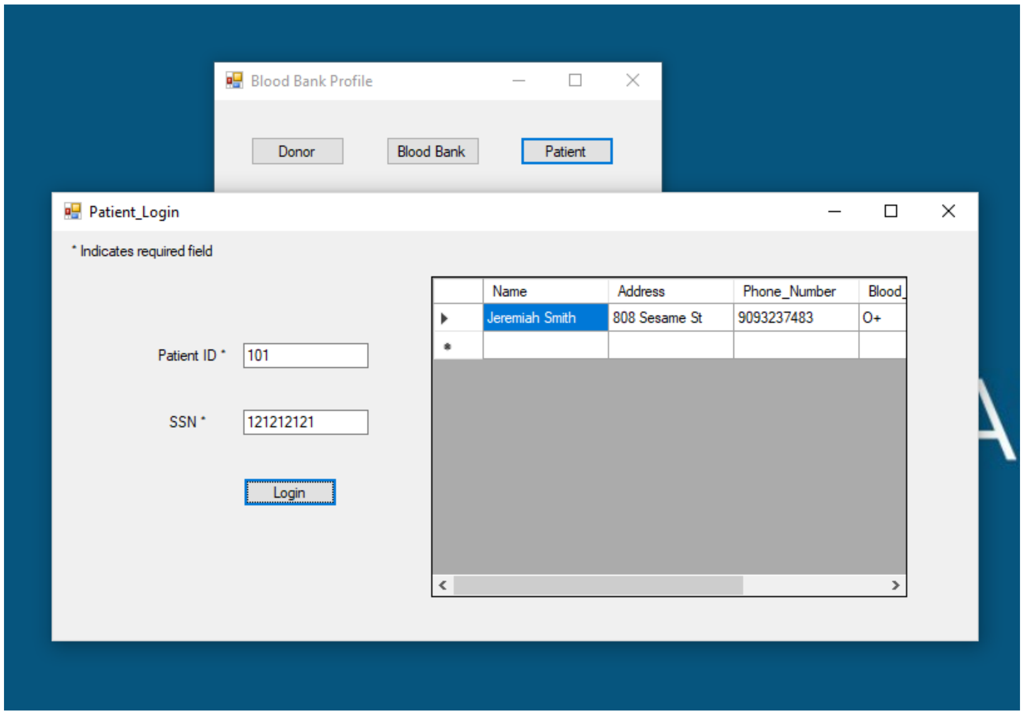
	Name	A		Blood_Type	Medical_History
	Kennedy Young	78		AB+	pregnant
	Lei He	80		B-	
	Lily Jacobs	678 Kensico St	9084328743	O-	
	Lyndsay Goldstein	787 I St	7839238484	A-	anxiety
	Mary Jane	78 Oak Lane	9004392323	AB+	brain bleed
	Sarah Smith	1900 F St NW	4672938989	B-	diabetes
	Sasha Smolyansky	469 Merriweather...	3920483029	O+	
	Stephen Morris	15 Pine St	7817295431	AB-	
	Warby Parker	56 Beech Tree R...	6009871234	A-	blind
	Yevgeny Smolya...	263 Berwind Road	6103899547	O-	
*					

Sort Profiles | Count Blood Types



Blood Bank Name

Name	Address	Phone_Number	Blood_Type	Medical_History
John Hudson	10 Ocean Drive	3657646732	AB+	HIV
John Snow	1 North Winter St	1234567890	O-	alive
Joseph Parks	4 23rd St	9098473241	O+	
Julia Roberts	23 Beverly Hills Dr	5637281234	AB-	HPV
Karen Mani	123 Cherry Lane	2027857347	AB+	hemophilia
Kennedy Young	78 A St	9087789672	AB+	pregnant
Lei He	800 21 St NW	5739294574	B-	
Lily Jacobs	678 Kensico St	9084328743	O-	
Lyndsay Goldstein	787 I St	7839238484	A-	anxiety
Mary Jane	78 Oak Lane	9004392323	AB+	brain bleed
Sansa Stark	1 Wall St	2345678901	AB+	
Sarah Smith	1900 E St NW	4672938989	B-	diabetes

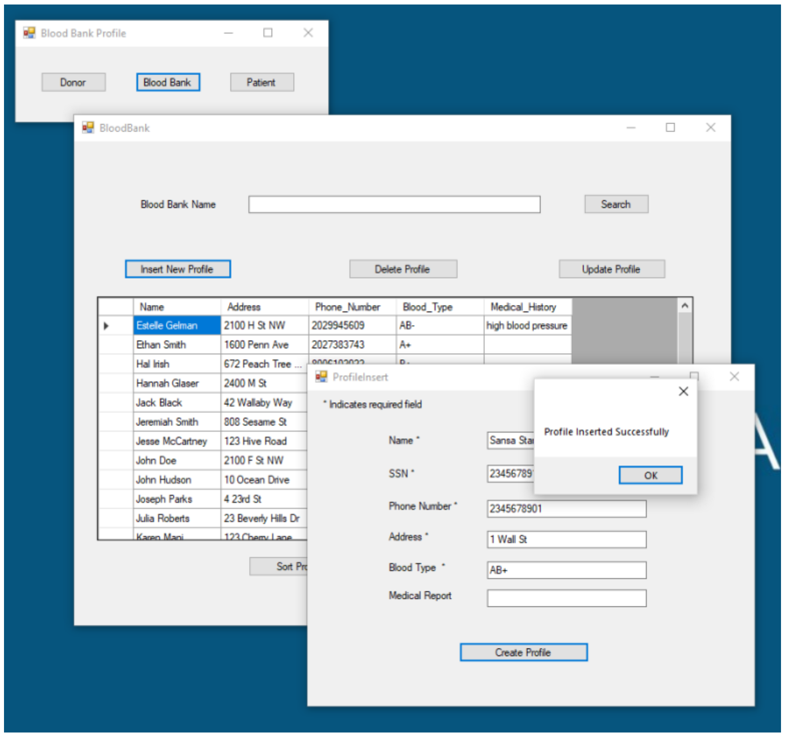


* Indicates required field

Patient ID * 101

SSN * 121212121

Name	Address	Phone_Number	Blood_
Jeremiah Smith	808 Sesame St	9093237483	O+


Blood Bank Profile

Donor | Blood Bank | Patient

BloodBank

Blood Bank Name | Search

Insert New Profile | Delete Profile | Update Profile

Name	Address	Phone_Number	Blood_Type	Medical_History
Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
Ethan Smith	1600 Penn Ave	2027383743	A+	
Hal Irish	672 Peach Tree ...	[illegible]	[illegible]	
Hannah Glaser	2400 M St			
Jack Black	42 Wallaby Way			
Jeremiah Smith	808 Sesame St			
Jesse McCartney	123 Hive Road			
John Doe	2100 F St NW			
John Hudson	10 Ocean Drive			
Joseph Parks	4 23rd St			
Julia Roberts	23 Beverly Hills Dr			
Karen Mani	123 Cherry Lane			

Sort Pr

ProfileInsert

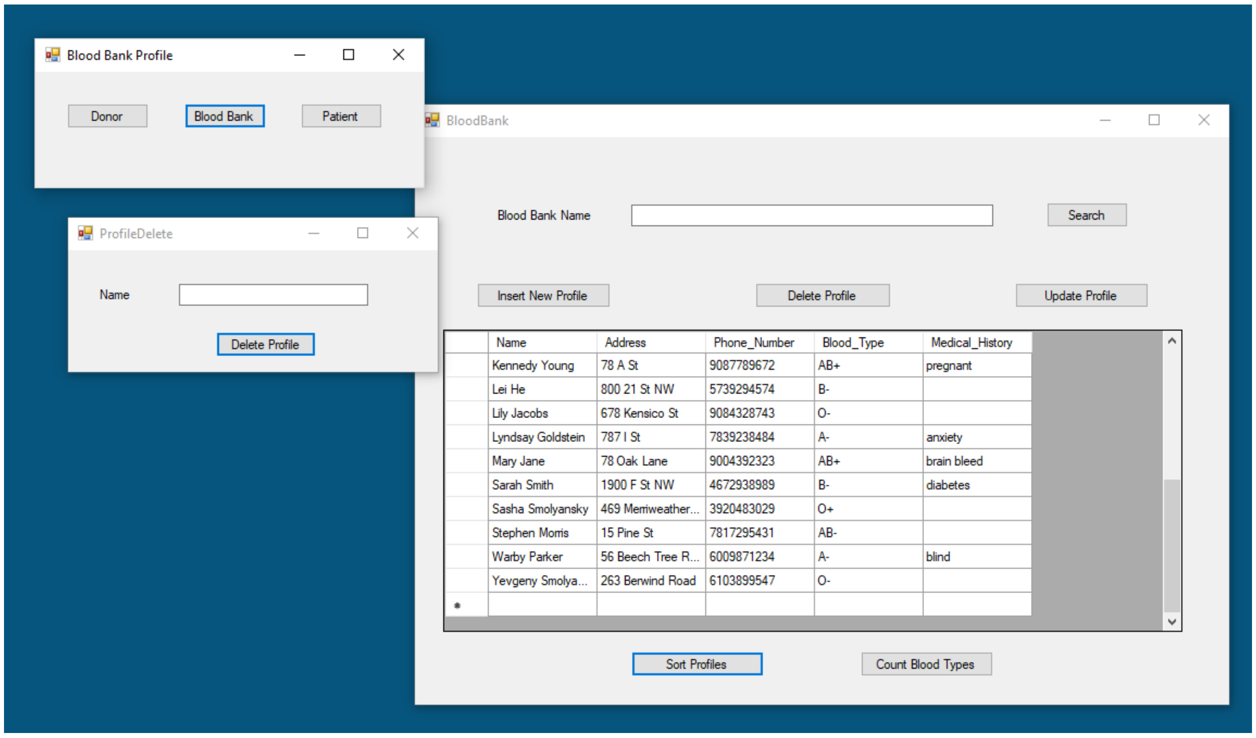
* Indicates required field

Name *	Sansa Sta
SSN *	23456789
Phone Number *	2345678901
Address *	1 Wall St
Blood Type *	AB+
Medical Report	

Create Profile

Profile Inserted Successfully

OK


Blood Bank Profile

Donor | Blood Bank | Patient

ProfileDelete

Name

Delete Profile

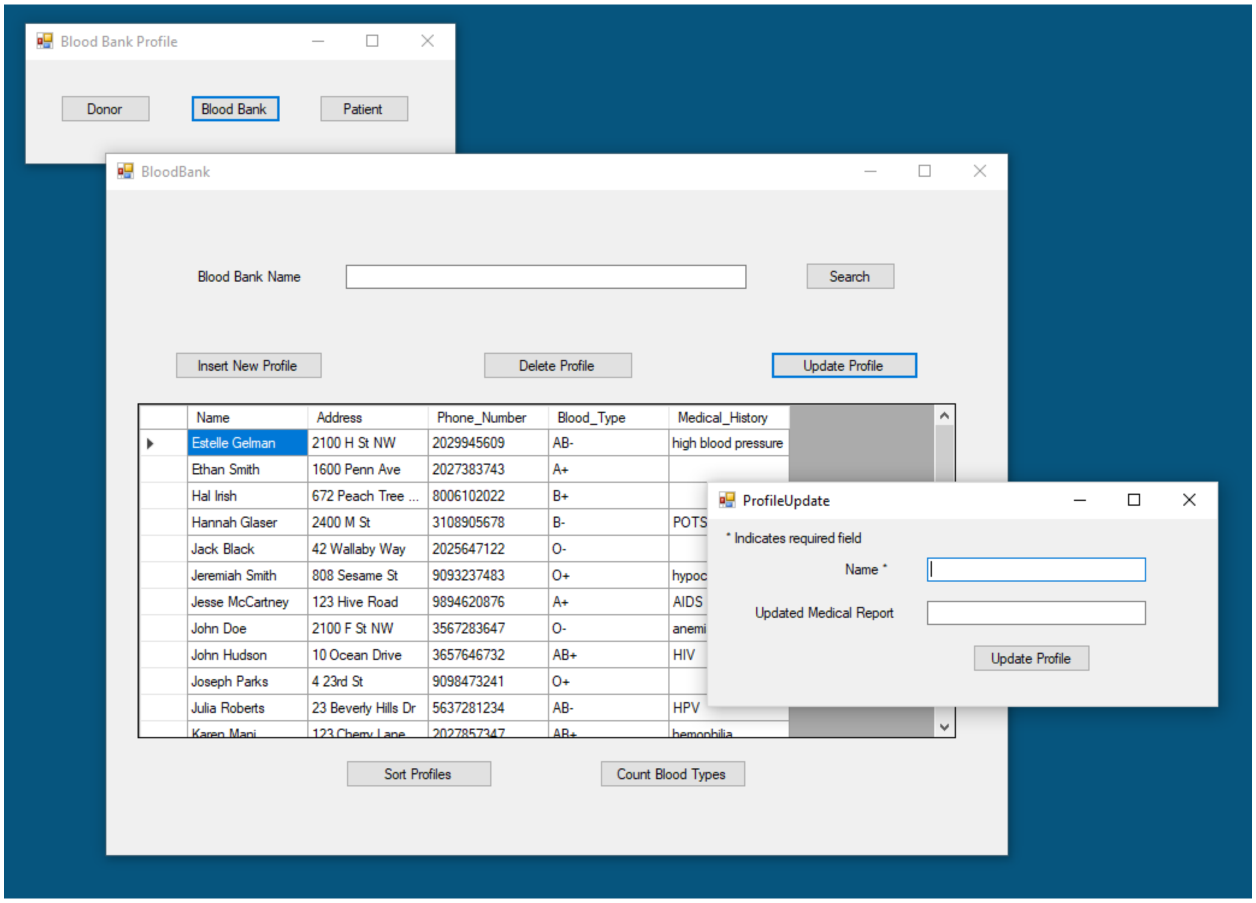
BloodBank

Blood Bank Name | Search

Insert New Profile | Delete Profile | Update Profile

Name	Address	Phone_Number	Blood_Type	Medical_History
Kennedy Young	78 A St	9087789672	AB+	pregnant
Lei He	800 21 St NW	5739294574	B-	
Lily Jacobs	678 Kensico St	9084328743	O-	
Lyndsay Goldstein	787 I St	7839238484	A-	anxiety
Mary Jane	78 Oak Lane	9004392323	AB+	brain bleed
Sarah Smith	1900 F St NW	4672938989	B-	diabetes
Sasha Smolyansky	469 Merriweather...	3920483029	O+	
Stephen Morris	15 Pine St	7817295431	AB-	
Warby Parker	56 Beech Tree R...	6009871234	A-	blind
Yevgeny Smolya...	263 Berwind Road	6103899547	O-	
*				

Sort Profiles | Count Blood Types


Blood Bank Profile

Donor | Blood Bank | Patient

BloodBank

Blood Bank Name [ ] Search

Insert New Profile | Delete Profile | Update Profile

	Name	Address	Phone_Number	Blood_Type	Medical_History
▸	Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
	Ethan Smith	1600 Penn Ave	2027383743	A+	
	Hal Irish	672 Peach Tree ...	8006102022	B+	
	Hannah Glaser	2400 M St	3108905678	B-	POTS
	Jack Black	42 Wallaby Way	2025647122	O-	
	Jeremiah Smith	808 Sesame St	9093237483	O+	hypoc
	Jesse McCartney	123 Hive Road	9894620876	A+	AIDS
	John Doe	2100 F St NW	3567283647	O-	anemi
	John Hudson	10 Ocean Drive	3657646732	AB+	HIV
	Joseph Parks	4 23rd St	9098473241	O+	
	Julia Roberts	23 Beverly Hills Dr	5637281234	AB-	HPV
	Karen Mani	123 Cherry Lane	2027857347	AB+	hemophilia

Sort Profiles | Count Blood Types

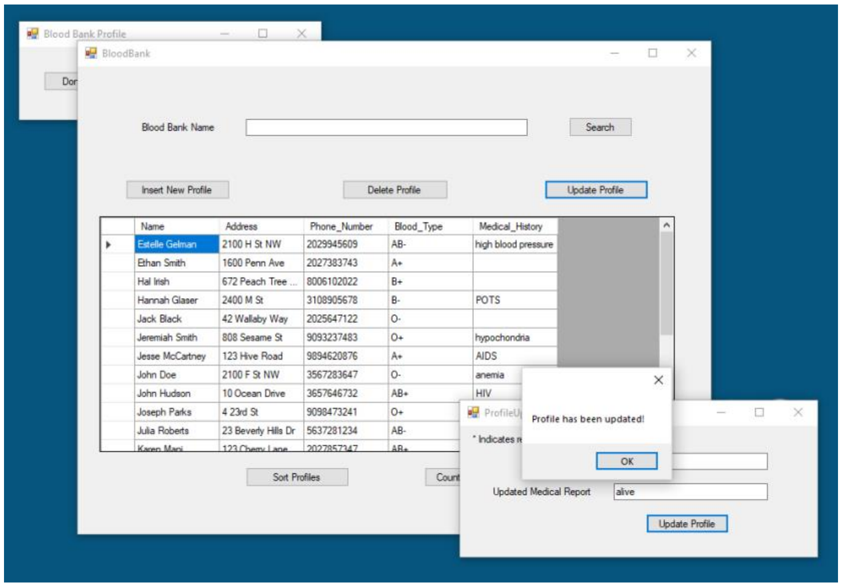
ProfileUpdate

* Indicates required field

Name * [ ]

Updated Medical Report [ ]

Update Profile


Blood Bank Profile

BloodBank

Blood Bank Name [ ] Search

Insert New Profile | Delete Profile | Update Profile

	Name	Address	Phone_Number	Blood_Type	Medical_History
▸	Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
	Ethan Smith	1600 Penn Ave	2027383743	A+	
	Hal Irish	672 Peach Tree ...	8006102022	B+	
	Hannah Glaser	2400 M St	3108905678	B-	POTS
	Jack Black	42 Wallaby Way	2025647122	O-	
	Jeremiah Smith	808 Sesame St	9093237483	O+	hypochondria
	Jesse McCartney	123 Hive Road	9894620876	A+	AIDS
	John Doe	2100 F St NW	3567283647	O-	anemia
	John Hudson	10 Ocean Drive	3657646732	AB+	HIV
	Joseph Parks	4 23rd St	9098473241	O+	
	Julia Roberts	23 Beverly Hills Dr	5637281234	AB-	
	Karen Mani	123 Cherry Lane	2027857347	AB+	

Sort Profiles | Count

ProfileU

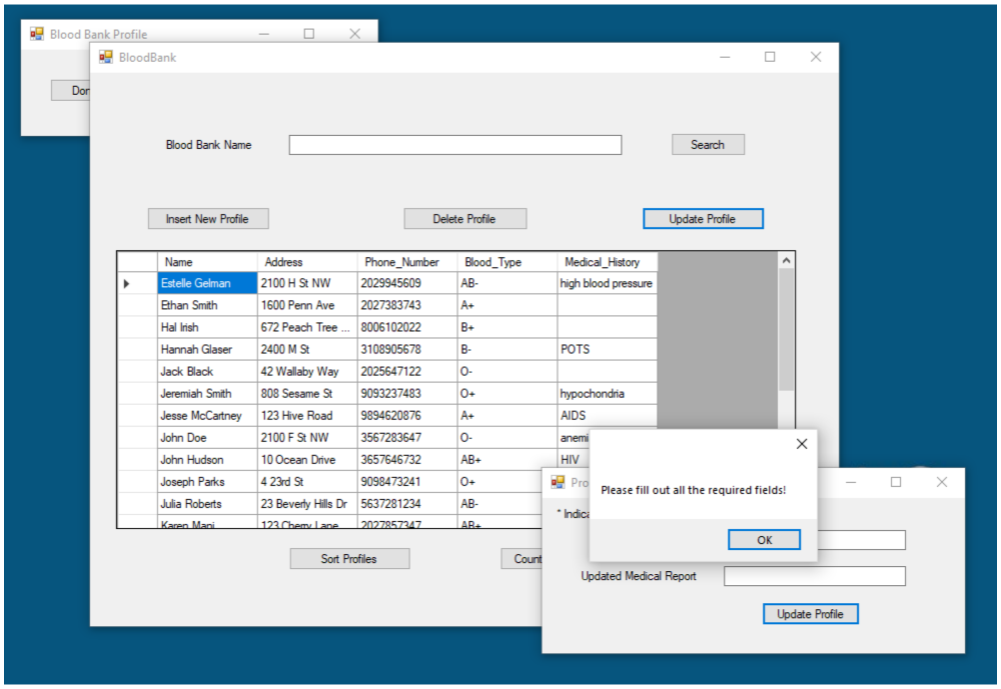
* Indicates r

Updated Medical Report [alive]

Update Profile

Profile has been updated!

OK



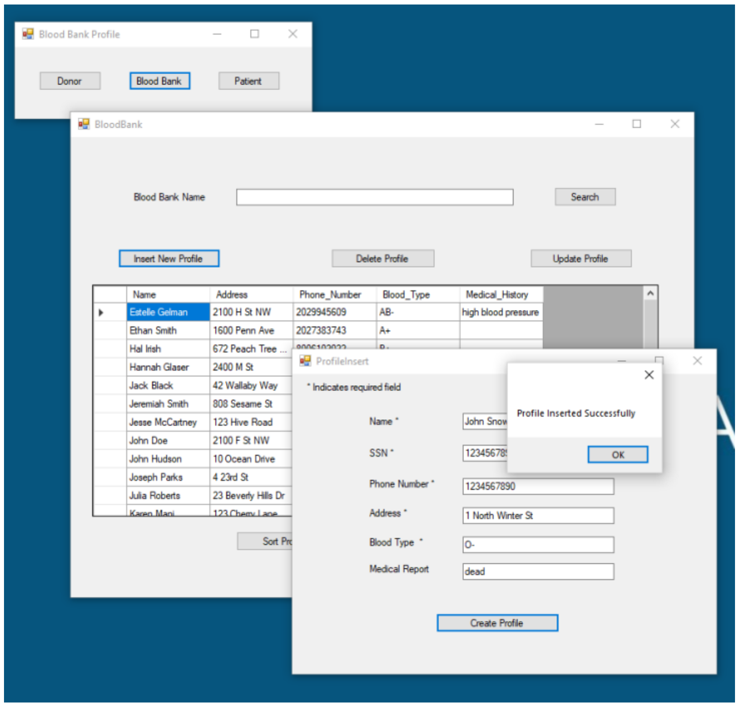
Blood Bank Name

	Name	Address	Phone_Number	Blood_Type	Medical_History
▶	Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
	Ethan Smith	1600 Penn Ave	2027383743	A+	
	Hal Irish	672 Peach Tree ...	8006102022	B+	
	Hannah Glaser	2400 M St	3108905678	B-	POTS
	Jack Black	42 Wallaby Way	2025647122	O-	
	Jeremiah Smith	808 Sesame St	9093237483	O+	hypochondria
	Jesse McCartney	123 Hive Road	9894620876	A+	AIDS
	John Doe	2100 F St NW	3567283647	O-	anemi
	John Hudson	10 Ocean Drive	3657646732	AB+	HIV
	Joseph Parks	4 23rd St	9098473241	O+	
	Julia Roberts	23 Beverly Hills Dr	5637281234	AB-	
	Karen Mani	123 Cherry Lane	2027857347	AB+	

Please fill out all the required fields!

* Indica

Updated Medical Report



Blood Bank Name

	Name	Address	Phone_Number	Blood_Type	Medical_History
▶	Estelle Gelman	2100 H St NW	2029945609	AB-	high blood pressure
	Ethan Smith	1600 Penn Ave	2027383743	A+	
	Hal Irish	672 Peach Tree ...			
	Hannah Glaser	2400 M St			
	Jack Black	42 Wallaby Way			
	Jeremiah Smith	808 Sesame St			
	Jesse McCartney	123 Hive Road			
	John Doe	2100 F St NW			
	John Hudson	10 Ocean Drive			
	Joseph Parks	4 23rd St			
	Julia Roberts	23 Beverly Hills Dr			
	Karen Mani	123 Cherry Lane			

* Indicates required field

Name *	John Snow
SSN *	12345678
Phone Number *	1234567890
Address *	1 North Winter St
Blood Type *	O-
Medical Report	dead

Profile Inserted Successfully

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.